Intelligent Electronic Navigational Aids: A New Approach

Costin Barbu, Maura Lohrenz and Geary Layne
Naval Research Laboratory
1005 Balch Blvd
Stennis Space Center, MS, 39529, USA
{cbarbu,mlohrenz,glayne}@nrlssc.navy.mil

Abstract—The smart management of clutter is a key component in designing intelligent, next-generation user interfaces and electronic displays. Intelligent devices can enhance a user’s situational awareness under adverse conditions. In this paper we present two approaches to assist a user with target detection and clutter analysis, and we suggest how these tools could be integrated with an electronic chart system. The first tool, an information fusion technique, is a multiple-view generalization of AdaBoost, which can assist a user in finding a target partially obscured by display clutter. The second technique clusters geospatial features on an electronic display and determines a meaningful measure of display clutter. The clutter metric correlates with preliminary, subjective, clutter rankings. The metric can be used to warn a user if display clutter is a potential hazard for his performance. We compare the performance of the proposed techniques with recent classifier fusion strategies on synthetic and real data.

I. INTRODUCTION

Over fifteen years ago, the US Navy first installed moving-map displays in the F/A-18 Hornet and AV-8B Harrier aircraft. Electronic charts are now commonplace in military and commercial aircraft, surface ships, and automobiles, and have proven essential to anyone needing immediate access to up-to-date geospatial information, such as meteorologists and air traffic controllers. As new sources of information become available for display, and as new and innovative display techniques are developed, there is a tendency to display everything that might be of interest to the user. These new displays introduce potential human factors’ issues with regard to the ability of the user to access and interpret the displayed information. Many studies have linked display complexity to user performance; e.g., display clutter has been shown to significantly disrupt a pilot’s visual attention, resulting in greater uncertainty concerning target locations [1], [18], [19]. When a moving-map scrolls at a high rate of speed, as in a fighter jet’s cockpit display, the chart’s effectiveness can decrease substantially. While researchers have demonstrated a link between user performance and the presence of so-called “clutter” (which can include both the overcrowding of otherwise important information as well as unwanted data or noise), we still lack a reliable method of automatically quantifying display clutter in a way that can be empirically tied to performance.

We illustrate the concept of information fusion employed by the first tool via a simple example. Given a set of training points $X = \{x_1, x_2, ..., x_N\}$ and $M$ disjoint features available for each point $x_i = \{x_i^1, x_i^2, ..., x_i^M\}$ (1)

Each member $x_i^j$ in the set $x_i$ is known as a view of point $x_i$. A view may be thought of as a representation of point $x_i$ using disjoint feature sets. For instance, in a color image, each training point $x_i$ may be thought of as a set of three views, each of which consists of one of the three disjoint features obtained from the intensities of Red, Green and Blue color components. In this case, the number of views will be three, represented as $\{x_i^R, x_i^G, x_i^B\}$. Similarly, for a moving target captured using visible range and infrared sensors, the number of views available for each training point in the training set will be two.

The goal of classifier fusion is to obtain a classifier $C$ such that $C$ learns from all the views available for each training point and has classification accuracy that is better than the case when only one view is available. One can ask how helpful could introducing additional views be? A toy example can be used to illustrate this concept. In Fig. 1 (a and b), two classes (circles and squares) are displayed on the OX and OY axes. It is not always possible to separate the classes using information from a single view. On the other hand, if information from all the views is combined, a better classification performance may be achieved. It is generally known that a good fusion algorithm outperforms or at least performs as well as the individual classifiers [14]. Considerable research in the pattern recognition field is focused on fusion rules that aggregate the outputs of the first level experts and make a final decision. Various techniques for fusion of expert observations such as linear weighted voting, the naive Bayes classifiers, the kernel function approach, potential functions, decision trees or multilayer perceptrons have been proposed in recent years, [9]. Other approaches are based on bagging, boosting, and arching classifiers [4], [5]. Comprehensive surveys of various classifier fusion studies and approached can be found in [10] and [11]. In [11] various classifier fusion strategies such as minimum, maximum, average, majority vote and oracle are discussed and the results have been compared. Kuncheva et al. [12] discuss the effect of dependence between individual classifiers in classifier fusion. They study the limits on the majority vote accuracy when combining dependent classifiers. A Q statistics based measure has been proposed to quantify the dependence between the classifiers.
## Intelligent Electronic Navigational Aids: A New Approach

### Authors
Naval Research Laboratory, 1005 Balch Blvd, Stennis Space Center, MS, 39529

### Abstract

### Distribution/Availability Statement
Approved for public release; distribution unlimited

### Security Classification
- a. Report: Unclassified
- b. Abstract: Unclassified
- c. This Page: Unclassified
It is shown that dependent classifiers could offer a dramatic improvement over the individual accuracy. A synthetic experiment demonstrates the intuitive result that, in general, negative dependence is preferable. In [20] Wolpert proposes stacked generalization, a general technique for construction of multi-level learning systems. In the context of classifier combination, it yields unbiased, full-size training sets for the trainable combiner. He defines stacked generalization as any scheme that feeds the information from one set of classifiers (generalizers) to another before forming the final opinion. Lanckriet et al. introduce in [13] a kernel-based data fusion approach for protein function prediction in yeast. The method presented in that paper combines multiple kernel representations in an optimal fashion by formulating the problem as a convex optimization problem that can be solved using semidefinite programming techniques.

In this paper, we present two tools that can be integrated with Intelligent Electronic Navigational Devices such that a user can be assisted when display clutter disrupts his visual attention. The first tool, a classifier fusion technique, is detailed in [3] and [2]. For the sake of clarity that fusion algorithm is briefly described in the next section. The second tool is a feature clustering-based technique that analyzes the tool is a feature clustering-based technique that analyzes the

Algorithm 1: Boosting with Shared Sampling Distribution (BSSD)

Input:
1. N training examples in a training set S.
2. M views available for each training point and hence M training sets such that $S_j = \{(x_{1j}^1,y_1), (x_{2j}^2,y_2), ..., (x_{Mj}^M,y_M)\}$ where $j = 1,...,M$ and $y_i \in \{+1,-1\}$ and each $(x_{ij}^j,y)$ pair represents the $j^{th}$ view and class label of the $i^{th}$ training example.

Initialization: The weights of the training examples are initialized to $w_1(i) = \frac{1}{N}$.

For $k = 1$ to $k_{\text{max}}$
1. For each view $j$, train classifiers $C_k^j$, using $W_k$
2. Obtain weak hypotheses $h_k^j$, for each view $j$
3. Obtain the error rates $\epsilon_k^j$ of each $h_k^j$ over the distribution $W_k$ such that $\epsilon_k^j = \frac{1}{w_k^j} \cdot \sum_{x_i \in S_k^j} I(h_k^j(x_i) \neq y_i)$
4. If errors from each of the $M$ views, $\{\epsilon_1^k, \epsilon_2^k, ..., \epsilon_M^k\} < 0.5$, select $h_k^*$ with the lowest error rate $\epsilon_k^*$ amongst all views
5. Compute the value of $\alpha_k^j = \frac{1}{\ln(1-\epsilon_k^j)}$ where $\epsilon_k^j = \min(\epsilon_1^k, \epsilon_2^k, ..., \epsilon_M^k)$ and $\alpha_k^j$ is the corresponding combination weight value.
6. Update the weights
$$w_{k+1}(i) = \frac{w_k(i)}{Z_k} \times \left\{ \begin{array}{ll}
e^{-\alpha_k^j} & \text{if } h_k^j(x_i^*) = y_i \\
e^{\alpha_k^j} & \text{if } h_k^j(x_i^*) \neq y_i \end{array} \right.$$

where $h_k^*$ is the classifier with lowest error rate $\epsilon_k^*$ in the $k^{th}$ iteration. $Z_k$ is the normalizing factor so that $W_{k+1}$ is a distribution.

Output: $F(x) = \sum_{k=1}^{k_{\text{max}}} \alpha_k^j h_k^j(x)$

Final hypothesis: $H(x) = \text{sign}(F(x))$

In the initialization step of Algorithm 1, all the views for a given training point are initialized with the same weight. To understand this we go back to the RGB component example. Suppose we have N training examples each having three disjoint views such that a given training example $x$ can be represented as $x = \{x^R, x^G, x^B\}$. Weak learners $h^R$, $h^G$ and $h^B$ will be trained on the training sets $X^R = \{x'^R_1, x'^R_2, ..., x'^R_N\}$, $X^G = \{x'^G_1, x'^G_2, ..., x'^G_N\}$ and $X^B = \{x'^B_1, x'^B_2, ..., x'^B_N\}$ such that $X = X^R \cup X^G \cup X^B$. Since the sampling distribution for all views of a given example is shared, the sampling weight of the the R, G and B views of example $x_k$ in iteration $k$ are given by
$$w_{k}^{R,G,B}(i) = w_k^R(i) = w_k^G(i) = w_k^B(i).$$

After a classifier $h_k^j$ with lowest error rate $\epsilon_k^j$ is selected in step 4 of Algorithm 1 and combination weight $\alpha_k^j$ is obtained, the weights of the views are updated.
III. QUANTIFYING VISUAL CLUTTER VIA FEATURE CLUSTERING

Previous studies on clutter (e.g. [15] and [21]) focus primarily on the contribution of saliency to image clutter. We theorize our perception of clutter is related to both saliency and color uniformity, or "density". Saliency refers to how clearly one color or feature "pops out" from the surrounding features in an image, which we estimate by a weighted average of color gradients between adjacent features. Color uniformity refers to how densely-packed are similarly-colored pixels within the image. To calculate this value, we have adapted a clustering algorithm, which we originally developed to cluster seafloor objects detected in sidescan sonar imagery. The algorithm clusters features detected within a predetermined geospatial distance from each other, produces vertices for a bounding cluster polygon, and calculates the cluster’s density as the number of clustered features divided by the area of the polygon. For this project, we adapted the clustering algorithm to operate in three-dimensional (3D) space, in which the third dimension is color. Our "color uniformity" value is then derived from the density of similarly-colored pixels within a 3D cluster (i.e., density = a weighted number of points within the cluster divided by the cluster's volume). We describe image clutter in terms of both local and global clutter components. A Local Clutter Metric (LCM) represents the contribution of one color or feature to the overall image clutter, and equals 1 minus the weighted average (by area) of the densities of all clusters centered on that color or feature. A Global Clutter Metric (GCM) represents the overall image clutter, equal to the weighted average of the LCM's for all colors or features in the image. Fig. 2 illustrates our proposed clutter function, in terms of saliency and LCM/GCM. The following sections describe in more detail how each of these metrics is calculated.

A. 3D Clustering using Geospatial Bitmaps (GB)

The original clustering algorithm relies on a geospatial bitmapping (GB) technique patented by NRL in 2001 [8]. The algorithm is unique in that it is an autonomous, consistently repeatable, computationally efficient "single-pass" method operating on a user-defined area of interest [7]. The algorithm clusters features by geospatial location and calculates a numerical measure of "cluster density" that considers the number and size of objects clustered in a given area, as well as the scale or resolution of the complete dataset. An enhancement to the original algorithm for this project is the ability to cluster features in three or more dimensions: two geospatial (x, y) dimensions plus a third (z) dimension such as color, size, or feature type. This paper presents preliminary results of clustering by geospatial location and color.

The GB clustering algorithm is a nonhierarchical algorithm with results similar to Nearest Neighbor (NN). NN iteratively calculates and compares the distances between every pair of elements in the dataset to determine which elements should be clustered together. In contrast, the GB algorithm is non-iterative, faster, less computationally intensive, and requires less computer memory than NN. The authors suggest that the GB algorithm is well suited to autonomous clustering applications, because the ordering of elements input to the algorithm has no effect on the resulting clusters (unlike NN and other single-pass methods), and the GB algorithm does not require a seed point to initiate clustering (unlike K-means and other relocation methods). The GB algorithm uses simple bitmaps, in which bits are turned on (set = 1) or off (cleared = 0), indicating the presence or absence of elements of interest. The index of each bit is unique and denotes its position relative to the other bits in the bitmap. In a 2D bitmap, each bit is indexed by its column (x) and row (y); in 3D, each bit is indexed by x, y, and depth (z). Although a GB can be defined for an entire finite space, memory is only allocated - dynamically - when groups of spatially close bits are set, resulting in a compact data structure that supports very fast Boolean and morphological operations. For this project, 3D bitmaps were used to cluster the pixels in an image of interest, based on geospatial location (x, y) and color (z). A separate clustering was performed for each color in the image. For example, Fig. 3 illustrates the results of clustering the shoreline pixels (darker brown color) in the sample image (left). All pixels within a geospatial distance of 1 (x and y) and a color distance of 9 (using the Commission Internationale d’Eclairage (CIE) L*a*b* color space) are included in the clusters (right). In this case, the resulting clusters only contain the shoreline pixels themselves. If z were increased to 10, every pixel in this image would be contained in a single cluster, because every pixel in this image is immediately surrounded by pixels that are within a color distance of 10 in CIE L*a*b* space.

B. Calculating Cluster Density

After clustering all pixels in the image into bounded polygons for a given "seed color" s, a cluster density $D_P$ is calculated for each cluster polygon $P$:
where: $D_S = Weighted\ average\ of\ clutter\ densities\ for\ all\ clusters\ centered\ on\ color\ s\ (described\ above)$

$A_1 = Sum\ of\ all\ areas\ A_S\ for\ image\ I$.

### D. Saliency

We estimate the local saliency of a given color or feature as a weighted average of the color differences between each color or feature of interest and immediately adjacent colors or features. For example, if one feature in the image (e.g., a yellow lighthouse symbol on a nautical chart) is completely surrounded by another feature (e.g., solid blue water), we would estimate the saliency of the lighthouse as the Euclidean distance between these two colors (yellow and blue) in a perceptually representative color space. If this lighthouse symbol were placed on a shoreline (brown), such that 40% of the lighthouse symbol was bordered by the blue water, 40% by tan land, and 20% by the brown shoreline, we would estimate the saliency of the lighthouse by $0.4 \ast (blue - yellow) + 0.4 \ast (tan - yellow) + 0.2 \ast (brown - yellow)$. Global saliency is estimated as the weighted average of the local saliencies for all colors (or features) in the image. Greater color distances result in greater saliency.

The choice of an appropriate color space is central to this theory. Unfortunately, no single color space has been shown to perfectly model human visual perception. For this paper, we chose the standard CIE L*a*b* color space, but we continue to search for improved options.

### IV. Experimental Results

We employed our fusion algorithm for target/clutter discrimination on two sets of binary class synthetic data and on a set of real data. We generated 32 target class images for each of the synthetic data sets such that a HUD (head-up display) symbol, a Bray-style flight path marker is included in each image as in [21]. The clutter class images are represented for the two synthetic data sets by images that share a common texture pattern. Sample images from both classes for the synthetic and real data sets are illustrated in Fig. 4. We consider the fusion of three views represented by the principal component projections, edges and wavelet coefficients for each image.

**Fig. 4.** Sample images of target (first row) and clutter (second row)

We empirically compare BSSD with the fusion methods stacked generalization (stacking), semidefinite programing (SDP/SVM) and majority vote (SVM–MV). Experimental results are presented in Tables I thru VI. The results represent the average accuracy of 20 tests, each time the data sets being randomly partitioned such that 60% of the data is in the training set and the remaining 40% is in
the test set. Average accuracy of an individual classifier from each view before fusion is shown in columns $A_{V_1}$, $A_{V_2}$ and $A_{V_3}$. The average fusion accuracy is presented in column $A_{fusion}$. Naive Bayes classifiers were used as weak learners for boosting. The SVM algorithm used as a back-end generalizer in stacking has two procedural parameters: $\sigma$ and $C$, the soft margin parameter. Ten-fold cross-validation was used for model selection, taking $\sigma$ values in $[10^{-2}, 10^2]$ and $C$ in $[10^{-2}, 10^2]$. Majority vote is also used for fusion of expert observations for the fusion techniques SVM–MV in which SVM has been used as classifier for each view. We used gaussian, polynomial and linear kernel functions on each view for the semidefinite programming technique. We compared the robustness of BSSD to noise with the competing techniques by randomly adding noise to the training data labels on all three views by flipping the labels.

We calculated global and local clutter metrics for the synthetic data (images with the target symbol surrounded by varying amounts of clutter vs. images with clutter only) and “real” data (aerial photographs of airport runways overlaid with HUD symbology vs. similar scenes without the HUD overlay). Results for the synthetic images are presented in Fig. 5; results for the real scenes are in Fig. 6. The synthetic images were binned into three groups ranging from lowest clutter (group 1) to highest clutter (group 3). To calculate local metrics for the no-target images, the darkest color of each image was chosen as the color of interest; for the target images, the target color (black) was chosen. The local clutter metrics (LCM and saliency) clearly delineated between synthetic images containing the target symbol and images containing only clutter. In general, images containing the target symbol exhibited higher local salience and lower local clutter than images without the target. The global metrics did not as clearly distinguish between the images containing the target and those without the target, since both sets of images contained equivalent amounts of background clutter.

Similarly, local clutter metrics clearly delineated between real airport scenes with HUD overlays and those without (in which pixel colors for the runways were used as the local feature of interest). In particular, the saliency of the HUD overlays was considerably higher than the saliency of the runways without HUD overlays. In addition, both local metrics (saliency and clutter) were significantly different than the global metrics for images with the HUD overlays, providing another cue for detecting this target. Conversely, local and global metrics were nearly identical for images without the HUD overlays. In other words, comparisons of both saliency and “color homogeneity” could be successfully used to predict how easily a HUD overlay might be detected (or how hard it might be to detect a runway without the HUD overlay) against various realistic background scenes.

V. SUMMARY AND DISCUSSION

In this paper we present two tools of potential utility to users of electronic chart displays. The first tool is a boosting-based classifier fusion that can assist a user in finding a
target when display clutter disrupts visual attention. The classifier fusion strategy performs classification using weak learners trained on different views of the training data. The final ensemble contains learners trained to focus on different views of the test data. The combination weights for the final weighting rule are obtained using a shared sampling distribution. In each iteration, one weak learner is selected from the pool of weak learners trained on disjoint views. This results in a minimization of the training error for the final hypothesis. It was shown in [3] that a lower training and generalization error bound can be achieved if a shared sampling distribution is used and a weak learner from the lowest error view is selected.

The second tool is a feature clustering technique that analyzes display clutter and attempts to determine whether a target of interest exists. Based on these analyses, the user could be warned by visual or acoustical alarms if his or her performance is likely to be affected by the amount of clutter in the display. The performance of the classifier fusion algorithm has been compared with other data fusion algorithms, namely stacking, majority vote and a semi-definite programming-based kernel method. We show that the proposed technique performs statistically significant better than other fusion techniques with > 95% confidence using a two-sided paired T-test.

### VI. ACKNOWLEDGMENTS

This work was sponsored under Program Element 002435N by the NRL 6.2 Base Program. The authors thank Dr. Marlin Gendron (NRL) for his valued assistance.

### REFERENCES


