Cue Integration through Discriminative Accumulation

M.E. Nilsback, B. Caputo

Computational Vision and Active Perception Laboratory (CVAP)
Dept. of Numerical Analysis and Computer Science
KTH, SE-100 44 Stockholm, Sweden

Abstract

Object recognition systems aiming to work in real world settings should use multiple cues in order to achieve robustness. We present a new cue integration scheme which extends the idea of cue accumulation to discriminative classifiers. We derive and test the scheme for Support Vector Machines (SVMs), but we also show that it is easily extendible to any large margin classifier. Interestingly, in the case of one-class SVMs, the scheme can be interpreted as a new class of Mercer kernels for multiple cues. Experimental comparison with a probabilistic accumulation scheme is favorable to our method. Comparison with voting scheme shows that our method may suffer as the number of object classes increases. Based on these results, we propose a recognition algorithm consisting of a decision tree where decisions at each node are taken using our accumulation scheme. Results obtained using this new algorithm compare very favorably to accumulation (both probabilistic and discriminative) and voting scheme.

1 Introduction

Recognizing objects on the basis of their visual appearance is one of the major goals in computer vision. This task has shown to be challenging, mainly because of the large variability in objects' appearance. Object categories vary considerably in their visual appearance, both between and within categories. The appearance of objects can change dramatically due to (self)occlusion, noise and different lighting conditions. Another well-known challenge for object recognition algorithms is the environment where the object is located. Indoor or outdoor scenes, and different types and degree of clutter, can considerably complicate the localization and recognition of an object. Most of research concentrates on building recognition algorithms relying on a single type of cue (see for instance [28, 16, 20, 25, 11] and many others). While these systems achieve remarkable performances for some applications, they are affected from some of the issues described above. A recognition system using a single, specific cue, assumes that this cue can be always detected and provide information sufficient for recognition. This assumption may be too strong for many cues, in many scenarios. For instance, global features like color or texture histograms tend to suffer from clutter and light changes. Local features are sensitive to view changes. Shape descriptors, by their very nature, do not handle occlusions well.

Many experiments on human visual perception (see [5] and the references therein) show that even humans perform poorly when artificially forced to use a single cue. This suggests that object recognition systems may achieve robustness by cue integration. Recognition with multiple cues is an important but somehow less researched issue in object recognition. Some authors have suggested building new representations that combine information derived from different cues. For example, Matas *et al* [17] proposed a new representation for objects with multiple colors, related to both histograms and region adjacency representations. Slater and Healey instead [24] suggested to use invariants of local color pixel distributions combined with the associated geometric information. Even if these type of features can achieve good performances for specific tasks, this kind of approach has two main drawbacks. First, combining more cues in a single feature vector is not likely to solve the robustness issues listed above. On the contrary, if one of the cues used gives misleading information, it is quite probable that the new feature vector will be adversely affected. Second, we can expect the dimension of such a feature vector to increase as the number of cues grows. This implies longer learning and recognition times, greater memory requirements and possibly curse of dimensionality effects. Another strategy is to use integration schemes [18, 26, 30]. Here, the pattern recognition literature offers a vast choice, but one of the most popular methods in object recognition is the voting scheme [15, 6, 12]. There are many possible variants of the voting scheme, but we can say that voting is, in general, dealing with a set of equivalent input cues and producing the output which is approved by most of them. Intuitively, voting relies on the probability that a majority of cues vote for a wrong hypothesis is lower than the probability that a minority of cues vote for the right hypothesis. This means that, for every decision, cues are divided into

reliable and *unreliable*. Then, the contribution of the unreliable cues is neglected and the decision is made based on the reliable cues. Thus, in cases when most of (or all) cues give wrong information, voting schemes are bound to misclassify the object.

In this paper we present a new integration scheme for multiple cues that does not neglect any cue contribution. We show with extensive experiments that this results in an algorithm that can perform correctly even when *all* cues indicate as the best hypothesis wrong answers. We start from the idea of cue integration via accumulation, introduced by Poggio et al [22] and Terzopoulos [29], and we extend it to discriminative classifiers. We focus on Support Vector Machines (SVMs) and we use the margin as the output of the classifier. For each cue, an SVM is trained and the corresponding margins are summed together (accumulated). Before summation, each margin is multiplied by a coefficient (learned from the training data) that indicates the reliability of that cue for the task at hand. The decision is made on the linear combination of all margins. Our scheme is naturally extendible to any margin-based classifier. We also show that, in the case of one-class SVM, it corresponds to defining a new class of Mercer kernels for multiple cues. We call this new cue integration method Discriminative Accumulation Scheme (DAS). We show that it can be used straightforwardly for recognition, or combined with a decision tree, where decisions at each leaf node are made using DAS.

The rest of the paper is organized as follows: Section 2 first discusses the general idea of accumulation and reviews the SVM algorithm (Section 2.1). It introduces DAS, it discusses how it can be extended to any margin-based classifier and it suggests a Mercer kernel interpretation of DAS for one-class SVMs (Section 2.2-2.3). Section 2 concludes with benchmark experiments of DAS against a probabilistic accumulation scheme. Section 3 compares DAS with voting schemes. We present a set of extensive experiments which show the strengths and weaknesses of both approaches. Motivated by these results, we propose a recognition algorithm consisting of a decision tree using DAS at each split node. Experiments with this new algorithm compare very favorably with probabilistic accumulation, voting scheme and DAS. The paper finishes with a summary discussion.

2 Integrating Cues via Accumulation

Many cue integration methods have been presented so far in the literature. For instance, Clark and Yuille [9] classify these methods into two main groups, *weak coupling* and *strong coupling*. Assuming that each cue is used as the input to a different classifier, weak coupling is when the output of two or more independent classifiers are combined.

On the other hand, strong coupling is when the output of one classifier is affected by the output of another classifier, so that their output are no longer independent. In this paper we focus on a weak coupling method called *accumulation*. The main idea of this method is that information from different cues can be summed together, thus accumulated. The idea was proposed first by Poggio et al [22] and Terzopoulos [29]; they accumulate cues by summing probabilities or by joint regularization. Probabilistic accumulation was further studied by Aloimonos and Shulman [2]. Here we extend the idea of accumulation to discriminative classifiers. Although we develop and test the scheme for SVMs, it is extendible to any margin-based classifier (such as Adaboost [13]). The rest of this Section presents our new method and reports benchmark experiments with a probabilistic accumulation scheme.

2.1 A Review on Support Vector Classifiers

Support Vector Machines (SVMs, [10, 31]) belong to the class of large margin classifiers. Consider the problem of separating the set of training data $(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)$, where $\boldsymbol{x}_i \in \Re^N$ is a feature vector and $y_i \in \{-1, +1\}$ its class label. If we assume that the two classes can be separated by a hyperplane $\boldsymbol{w} \cdot \boldsymbol{x} + b = 0$, and that we have no prior knowledge about the data distribution, then the optimal hyperplane is the one which has maximum distance to the closest points in the training set. The optimal values for $\boldsymbol{w}$ and b can be found by solving the following constrained minimization problem:

$$\underset{\boldsymbol{w}, b}{\text{minimize}} \quad \frac{1}{2} \|\boldsymbol{w}\|^2, \quad \text{subject to} \quad y_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + b) \geq 1,$$

$\forall i = 1, \ldots, m$. Solving it using Lagrange multipliers $\alpha_i (i = 1, \ldots, m)$ results in a classification function $f(x) = \text{sgn}\left(\sum \alpha_i y_i \boldsymbol{w} \cdot \boldsymbol{x} + b\right)$, where α_i and b are found by using an SVC learning algorithm [10, 31]. Most of the α_is take the value of zero; those $\boldsymbol{x}_i$ with nonzero α_i are the "support vectors" (for the extension to the non linearly separable case, we refer the reader to [10, 31] and the references therein). To obtain a nonlinear classifier, one maps the data from the input space $\Re^N$ to a high dimensional feature space $\mathcal{H}$ by $\boldsymbol{x} \to \Phi(\boldsymbol{x}) \in \mathcal{H}$, such that the mapped data points of the two classes are linearly separable in the feature space. Assuming there exists a kernel function K such that $K(\boldsymbol{x}, \boldsymbol{y}) = \Phi(\boldsymbol{x}) \cdot \Phi(\boldsymbol{y})$, then a nonlinear SVM can be constructed by replacing the inner product $\boldsymbol{x} \cdot \boldsymbol{y}$ in the linear SVM by the kernel function $K(\boldsymbol{x}, \boldsymbol{y})$, obtaining then $f(\boldsymbol{x}) = \text{sgn}\left(\sum_{i=1}^{m} \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b\right)$. This corresponds to constructing an optimal separating hyperplane in the feature space.

The extension of SVM to multi class problems can be done using the so called *one-vs-other* strategy. If M is the

number of classes, M SVM's are trained, each separating a single class from all remaining classes. The quantity of interest in this case is

$$D_j = \sum_{i=1}^{m_j} a_{ij} y_{ij} K(\boldsymbol{x}_{ij}, \boldsymbol{x}) + b_j$$

which is the margin. The final output of the classifier is

$$j^* = \operatorname*{argmax}_{j=1}^{M} \{D_j\},$$

the SVM with the highest output value.

SVMs can also be extended to one-class problems. The main idea is to find the sphere in feature space of minimum radius which contains most of the data of the training set. The possible presence of outliers is countered by using slack variables ξ_i which allow for data points outside the sphere. In this case the decision function is given by

$$f(\boldsymbol{x}) = \operatorname{sgn}\left(\sum_i \alpha_i K(\boldsymbol{x}, \boldsymbol{x}_i) - \sum_{ij} \alpha_i \alpha_j K(\boldsymbol{x}_i, \boldsymbol{x}_j)\right).$$

2.2 Discriminative Accumulation

Suppose we are given M object classes and, for each class, a set of N_j training images $\{\boldsymbol{I}_i^j\}_{i=1}^{N_j}, j = 1, \dots M$. Suppose also that, from each image, we extract a set of P different cues:

$$T_p = T_p(\boldsymbol{I}_i^j), \quad p = 1 \dots P.$$

The goal is to perform object recognition using all the cues. Our idea is to use an accumulation scheme based on the output of effective discriminative classifiers like SVMs. A key feature of cue integration via accumulation is that, even when most of the cues provide a wrong answer, the final classifier still has a chance to perform correctly, due to the accumulation effect (for an example we refer the reader to [7], and to results reported in Section 2.4). We expect, that combining the accumulation idea with the power of SVMs as single-cue classifiers, will make accumulation even more effective. Our new *Discriminative Accumulation Scheme (DAS)* can be described in two steps:

Step 1: Single-cue SVMs From the original training set $\{\boldsymbol{I}_i^j\}_{i=1}^{N_j}$, for each object j, with $j = 1, \dots, M$ define P new training sets $\{T_p(\boldsymbol{I}_i^j)\}_{i=1}^{N_j}, j = 1, \dots, M, p = 1 \dots, P$, each relative to a single cue. For each new training set we train an SVM. In general, kernel functions may differ from cue to cue. Model parameters can be estimated during the training step via cross validation. Then, given a test image $\widehat{\boldsymbol{I}}$ and assuming $M \geq 2$, for each single-cue SVM we compute the margin:

$$D_j(p) = \sum_{i=1}^{m_j^p} \alpha_{ij}^p y_{ij} K_p(T_p(\boldsymbol{I}_i^j), T_p(\widehat{\boldsymbol{I}})) + b_j^p.$$

The index p on $(m_j^p, \alpha_{ij}^p, K_p(\cdot, \cdot), b_j^p)$ indicates that in general these quantities have different values for different cues.

Step 2: Discriminative Accumulation After we collect all the margins $\{D_j(p)\}_{p=1}^{P}$, for all the j objects $j = 1, \dots, M$ and the p cues $p = 1, \dots, P$, we classify the image $\widehat{\boldsymbol{I}}$ using their linear combination:

$$j^* = \operatorname*{argmax}_{j=1}^{M}\left\{\sum_{p=1}^{P} a_p D_j(p)\right\}, a_p \in \Re^+. \quad (1)$$

$\{a_p\}_{p=1}^{P}$ are also evaluated via cross validation during the training step. This means that the relevance of each cue, for a specific task, is evaluated during the training step from the training data (see Sections 2.4 and 3 for examples). Eq. (1) is clearly an accumulation scheme because all contributions from all cues are summed together. At the same time, it is discriminative, in the sense that the contribution of each cue is obtained via SVMs. To the best of our knowledge, this is the first cue integration scheme that combines SVMs with the idea of accumulation.

The algorithm can be used also for $M = 1$ (object detection). The scheme is the same except that, in this case, the margin $D(p)$ is given by

$$\begin{aligned} D(p) &= \sum_{i=1}^{m^p} \alpha_i^p K_p(T_p(\boldsymbol{I}_i), T_p(\widehat{\boldsymbol{I}})) - \\ &\quad \sum_{i,k=1}^{m^p} \alpha_i^p \alpha_k^p K_p(T_p(\boldsymbol{I}_i), T_p(\boldsymbol{I}_k)). \end{aligned}$$

Note that in this case $D(p)$ is a linear combination of kernel functions. Thus Eq (1) can be interpreted as a one-class SVM using as kernel the function:

$$K_{MC}(\{T_p(\boldsymbol{I}_i)\}_p, \{T_p(\boldsymbol{I})\}_p) = \sum_{p=1}^{P} a_p K_p(T_p(\boldsymbol{I}_i), T_p(\boldsymbol{I})).$$

$K_{MC}(\{T_p(\boldsymbol{I}_i)\}_p, \{T_p(\boldsymbol{I})\}_p)$ is a multi-cue Mercer kernel, as it consists of a linear combination with positive coefficients of Mercer kernels [10].

2.3 Generalization to Margin-based Classifiers

The discriminative accumulation scheme described in Section 2.2 is based on the margins $\{D_j(p)\}$, which are the outputs of each single-cue SVM. It follows that the scheme can be used for *any* margin-based classifier. These are a class of learning algorithms which take as input binary labeled training examples $(\boldsymbol{x}_1, y_1), \dots (\boldsymbol{x}_m, y_m)$ with $\boldsymbol{x}_i \in \chi$ and $y_i \in \{+1, -1\}$ (for the extension to multi classes we refer the reader to [1] and the references therein). Data are used

	CH	MFH	CH-MFH
SVM	**16.18%**	**5.33%**	
SG-MRF	20.10%	6.28%	
DAS			**1.82%**
USG-MRF			3.55%

Table 1: Results for SVM, SG-MRF, DAS and USG-MRF on the COIL database. Results for SG-MRF and USG-MRF are reported from [7].

to generate a real-valued function or hypothesis $f : \chi \to \Re$, with f belonging to some hypothesis space F. The margin of an example $\boldsymbol{x}$ with respect to f is $f(\boldsymbol{x})$. $f(\boldsymbol{x})$ is determined by minimizing:

$$\frac{1}{m} \sum_{i=1}^{m} L(y_i f(\boldsymbol{x}_i)), \qquad (2)$$

for some loss function $L : \Re \to [0, \infty[$. The generalization of the discriminative accumulation scheme to any margin-based algorithm is straightforward: given a margin-based classifier, we will train P single-cue algorithms and compute for each of them the corresponding quantity $f_j(T_p(\boldsymbol{x})), p = 1, \ldots P$, relative to the object $j, \forall j = 1, \ldots M$ (step 1). The multi-cue classifier will become (step 2)

$$j^* = \underset{j}{\operatorname{argmax}} \left\{ \sum_{p=1}^{P} a_p f_j(T_p(\boldsymbol{x})) \right\}, a_p \in \Re^+.$$

Different choices of the loss function L and different algorithms for minimizing Eq (2) over some hypothesis space lead to various well-studied learning algorithms, such as SVMs, Adaboost ([13], which has proved to be very effective for object recognition [32]), regression and decision trees.

2.4   Results

In order to assess the effectiveness of DAS for object recognition, we ran a first set of benchmark experiments against a probabilistic accumulation scheme. It was recently proposed in [7] an integration scheme based on accumulation of kernel Gibbs distributions called Spin Glass-Markov Random Fields (SG-MRF). The authors derived the algorithm from results of statistical physics (hence the name of the model), but the final classifier can also be interpreted as a probabilistic accumulation scheme for multiple cues. This is (to our knowledge) the most recent example of probabilistic accumulation for object recognition, and also a kernel method. Thus, we decided to test our scheme against SG-MRFs on two different databases. We report results obtained by using single-cue SVMs and DAS; we decided not

DAS	**1st match**	**1st match**	**1st match**
SVM(CH)	**1st match**	52nd match	59th match
SVM(MFH)	35th match	**1st match**	2nd match

Table 2: COIL database experiments: examples of images misclassified by one or both cues, and classified correctly by DAS.

to run experiments on concatenated feature vectors, as done in [7], for two reasons: (1) results reported there suggest that it is not an effective strategy compared to accumulation. (2) The intrinsic weaknesses of multi-cue features for object recognition, that we discussed in Section 1.

COIL Database Experiments We repeated the experiment described in [7], which now we briefly summarize. We used the COIL database [19] with training set of 12 views per object (one every $30°$); training and test set were disjoint. As features, we used Color Histograms (CH [27], rg with resolution of bin axis equal to 8) and Multidimensional receptive Field Histograms (MFH [25], Gaussian derivative filters along x and y directions, $\sigma = 1.0$ and resolution of bin axis equal to 8). Both histograms were normalized to 1. We ran experiments with SVM on each cue separately, and with DAS on both cues. The kernel used was [31]

$$K(\boldsymbol{x}, \boldsymbol{y}) = \exp\left\{ -\gamma \sum_i ||x_i^a - y_i^a||^b \right\}, a \in \Re^+, b \in]0, 2].$$

Here, and for all the experiments reported in this paper, we used a modified version of LIBSVM [8], with $C = 100$. Experiments were run on a SUN Ultra-Enterprise with memory size of 4096 Megabytes, 8 Processors UltraSPARC-II 400Mhz. Results (error rates) are reported in Table 1. We see that, in all cases, our algorithm performs better than SG-MRF [1]. Learning time (= finding the support vectors plus model selection) went from a minimum of 11886 s for MFH to a maximum of 30047 s for DAS (the code is far from being optimized; also, cross validation for the kernel parameters (a, b, γ) has been quite time consuming). Recognition time per view went from a minimum of 28.3 ms using SVM and CH, to a maximum of 57.5 ms using DAS. The parameters $\{a_p\}_{p=1}^{2}$, found via cross validation, were 0.7 for CH and 1 for MFH.

Table 2 shows in details some examples of classification results. The left and middle columns show examples of object views misclassified by one of the two cues, but classified correctly by DAS. The right column shows an example

---

[1]Results reported in [7] relative to SVMs were obtained using a Gaussian kernel.

Figure 1: Examples of images from the Caltech database.

	CH	MFH	CH-MFH
SVM	**6.63%**	**3.11%**	
SG-MRF	11.93%	8.12%	
DAS			**1.55%**
USG-MRF			3.39%

Table 3: Results for SVM, SG-MRF, DAS and USG-MRF on the Caltech database; we report the error rates.

of object view misclassified by both cues, but classified correctly by DAS. This type of behavior is a major point in favor of accumulation: in such a case, a voting scheme would give a wrong answer. USG-MRF displays a similar behavior [7], but the error rates reported in Table 1 show clearly that DAS is more effective than USG-MRF in this case.

Caltech Database Experiments We ran a second set of experiments on images of cars (rear), airplanes, motorbikes and faces, taken in real world scenes and at different scales (Caltech database, [11]). Training and test set consists of 400 images each for cars, airplanes and motorbikes, 218 for training and 217 for test for faces. The experimental setup was analogous to the one described for the COIL experiment. The only differences were the choice for the feature parameters (rg and resolution of bin axis equal to 16 for CH; Laplacian derivative filters at two different scales ($\sigma_1 = 1.0, \sigma_2 = 2.0$) and resolution of bin axis equal to 16 for MFH), and the kernel used for SVM, which in these experiments was [4]

$$K(\boldsymbol{x}, \boldsymbol{y}) = \exp\left\{-\gamma \chi^2(\boldsymbol{x}, \boldsymbol{y})\right\}. \tag{3}$$

Results are reported in Table 3. Once again, we see that DAS performs better than SG-MRF. It is interesting to note the good performance achieved by both methods even using a single cue. The worst results (11.93% error rate, obtained by SG-MRF using the CH representation) is quite remarkable, considering that it is obtained with a global type of feature, on object images taken in cluttered backgrounds. Learning time for DAS went from a minimum of 452 s to a maximum of 1392 s (note the difference with the previous experiment; in this case, the only kernel parameter to be selected was γ). Recognition time per view went from a minimum of 6 ms using SVM and CH to a maximum of 18 ms using DAS. The parameters $\{a\}_{p=1}^2$, found via cross validation, were 1.1 for CH and 1 for MFH.

Discussion Both experiments show a clear improvement in

results by using the discriminative approach. Of course, it could be argued that SG-MRFs' performance is poorer because of the way the pdfs are estimated. It is possible that, by evaluating the pdfs differently, results for the probabilistic accumulation scheme would improve, and maybe surpass those presented for DAS. The truth is that both accumulation schemes (probabilistic and discriminative) have a heuristic element. For the probabilistic approach, it is how to estimate the pdfs; for the discriminative approach, it is how to choose the kernel function. In both cases, a bad choice can affect the final performance. Still, SVMs are gaining popularity for visual pattern recognition, and this has two consequences: first, there is an increasing literature reporting results on several applications, using different types of cues and different kernels ([23, 33, 14] and many others). This is building a shared knowledge regarding which kernel should be used for a given feature type. Second, as more new kernel functions are derived for specific visual recognition problems (see for instance [34, 35, 3]), it is possible to use more cue types for DAS. Thus, choosing a good kernel in designing an SVM-based algorithm is becoming less and less heuristic. This, united with the experimental results we have shown, makes us conclude in favor of DAS with respect to probabilistic accumulation schemes.

3 Discriminative Accumulation and Voting Scheme

Voting schemes are one of the most popular approaches for cue integration in object recognition [15, 6, 12]. The idea of integrating cues by voting can be implemented with many possible variations [21]. In this paper, we compare DAS with the voting scheme used in [15], which to our knowledge is the most recent example of object recognition with multiple cues integrated by a voting algorithm. Although there are, of course, many other possible different implementations for a voting algorithm, these experiments still give some indications of the strengths and weaknesses of these two approaches.

CogVis-ETH Database Experiments We ran a first experiment on the CogVis-ETH database [15], which contains 80 objects from 8 different categories (apple, tomato, pear, toy-cow, toy-horse, toy-dog, toy-car and cup). We designed the experiment to make our results comparable to those reported in [15]. We used two cues: (1) MFH with Gaussian derivative filters along x and y directions, three scales ($\sigma_{1,2,3} = 1.0, 2.0, 4.0$) and resolution of bin axis equal to 16; (2) CH with RGB, resolution of bin axis equal to 16. For each category, the training set consisted of 9 objects and test set of 1 object. We generated 10 different partitions of training and test set, and we ran accordingly 10 experiments (we present the averaged results). We used the kernel described

	CH	MFH	CH-MFH
Leibe-Schiele [15]	35.15%	20.21%	13.6%
SVM	**19.97%**	**6.07%**	
DAS			4.0%
SVM-Voting			**3.6%**

Table 4: Error rates for the CogVis-ETH database. Results in the first row are reported from [15], where the result in the last column is obtained with the voting scheme. The result in the third row is obtained by using a voting algorithm with single-cue SVMs as inputs.

in Eq (3). We compared results obtained using DAS with those reported in [15] [2], and with results obtained by using the voting scheme on the outputs of the one-cue SVMs. Results are reported in Table 4.

Two considerations must be made: first, SVMs with single cue give much better results than using χ^2 [15]. As a consequence, both cue integration schemes using as input SVMs perform quite better than the voting scheme on χ^2. Second, the voting scheme using single-cue SVMs as input performs slightly better than DAS (a 0.4% difference in error rate).

Caltech Database Experiments We ran a second experiment on the Caltech database. This time we used three different cues: MFH and CH with the feature parameters described for this database in Section 2.4, and jet features [28] consisting of 78 feature points, computed over 7 different scales, each resulting in a 9-dimensional vector. The kernel used for jet features was [34] $K_L(\boldsymbol{L}_h, \boldsymbol{L}_k) = 1/2[\widehat{K}(\boldsymbol{L}_h, \boldsymbol{L}_k) + \widehat{K}(\boldsymbol{L}_k, \boldsymbol{L}_h)]$, with

$$\widehat{K}(\boldsymbol{L}_h, \boldsymbol{L}_k) = \frac{1}{n_h} \sum_{j_h=1}^{n_h} \max_{j_k=1,\ldots,n_k} \{K_l(\boldsymbol{l}_{j_h}(\boldsymbol{L}_h), \boldsymbol{l}_{j_k}(\boldsymbol{L}_k))$$

were $\boldsymbol{L}_i = \{\boldsymbol{l}_j(\boldsymbol{I}_i)\}_{j=1}^{n_i}$ is a jet feature vector for an image $\boldsymbol{I}_i$ and K_l is

$$K_l(\boldsymbol{x}_{j_h}, \boldsymbol{y}_{j_k}) = \frac{\langle \boldsymbol{x}_{j_h} - \boldsymbol{\mu_x} | \boldsymbol{y}_{j_h} - \boldsymbol{\mu_y} \rangle}{||\boldsymbol{x}_{j_h} - \boldsymbol{\mu_x}|| \cdot ||\boldsymbol{y}_{j_h} - \boldsymbol{\mu_y}||}$$

Results obtained using DAS and the voting scheme on single-cue SVMs, for the three cues and for the (MFH-jet) and (MFH-CH) combinations are reported in Table 5, top and middle row (results for the (CH-jet) combination are analogous to those obtained with (MFH-jet), thus we skip them).

Here we observe a different behavior: for all cue combinations, DAS gives the better performance. We explain the results of these two experiments as follows: DAS is an accumulation scheme, hence uses all cues, for every decision.

---

[2]Although in [15] the authors defined training and test set differently, the statistical significance should be preserved.

Each cue is weighted by a coefficient a_p, which indicates somehow the degree of reliability of that cue. These coefficients are determined during the learning step, via cross validation, optimizing the *overall* performance. Thus, when the number of object classes grows, each cue will have to compromise more: the recognition rate of some objects might decrease slightly, to avoid that the recognition rate of all the remaining objects does not decrease too much.

A DAS-based Decision Tree The voting algorithm used is not affected by the increasing number of objects, because it is based on a decision tree which splits the object classes in two subgroups at each step, optimizing their recognition rate. How to perform the splitting is decided automatically by the algorithm itself (for more details, we refer the reader to [15]) . This consideration brings us to suggest a new integration scheme combining the best of both worlds, namely a decision tree where decisions are taken using DAS at each node leaf. In this way, DAS parameters $\{a_p\}$ would always be selected for a two-class problem. If our hypothesis on the behavior of DAS is correct, this should allow DAS to exploit fully its potential. We tested our idea by running the two experiments described in this Section, using the DAS-Decision Tree (DAS-DT). For the CogVis-ETH database we obtained an error rate of 2.89%, which is a 0.71% better than the voting scheme using single-cue SVMs, and a 1.11% better than using DAS (see Table 4 for the results on the CogVis-ETH database without using DAS-DT).

Results for the Caltech experiments are reported in Table 5, bottom row. We see that the behavior is similar to what observed for the CogVis-ETH experiments: DAS-DT obtains a better performance than DAS and voting scheme, for all possible different combinations of the three cues. A careful analysis of the results confirms our hypothesis on the performance of DAS for many object classes. For instance, consider in detail results obtained for DAS and DAS-DT, using MFH and jet features (Table 5, left column, middle and bottom row). We see that the error rate found by DAS for motorbikes is lower than the error rate found by DAS-DT for the same object class. Still, error rates for the other three objects decrease (except for cars, for which both methods obtain a 0% error rate), and this brings to an *overall* better performance for DAS-DT. It is interesting to note how, for a given experiment, the weighting coefficients $\{a_p\}$ change, for each cue, by using DAS or DAS-DT. For example, in the case of the MFH-CH experiment (Table 5, middle column, middle and bottom row), we see that DAS finds a 1.0 coefficient for MFH, and a 1.1 for CH. In the case of DAS-DT, the coefficient for MFH remains constant and equal to 1 for all splitting, while the coefficient for CH is found to be 1 for the first splitting, 2 for the second splitting and 3.5 for the third. Thus, using a tree structure allows the DAS decision rule to adapt the weighting of each feature at each split, which leads to a better overall performance.

Voting(MFH-jet)

MFH
MFH
MFH

Total Error Rate: **3.06 %**

| car | face | airplane | motorbike |
| 0% | 2.76% | 3.25% | 6.25% |

Voting(MFH-CH)

MFH
MFH
MFH

Total Error Rate: **3.06 %**

| car | face | airplane | motorbike |
| 0% | 2.76% | 3.25% | 6.25% |

Voting(MFH-CH-jet)

MFH
MFH
MFH

Total Error Rate: **3.06 %**

| car | face | airplane | motorbike |
| 0% | 2.76% | 3.25% | 6.25% |

DAS(MFH-jet)

$a_{MFH} = 1.00$
$a_{JET} = 1.39$

Total Error Rate: **1.17 %**

| car | face | airplane | motorbike |
| 0% | 0.92% | 1.25% | 2.50% |

DAS(MFH-CH)

$a_{MFH} = 1.00$
$a_{CH} = 1.10$

Total Error Rate: **1.44 %**

| car | face | airplane | motorbike |
| 0% | 0% | 2.00% | 3.50% |

DAS(MFH-CH-jet)

$a_{MFH} = 1.00$
$a_{CH} = 0.90$
$a_{JET} = 0.90$

Total Error Rate: **0.62 %**

| car | face | airplane | motorbike |
| 0% | 0% | 1.75% | 0.75% |

DAS-DT(MFH-jet)

$a_{MFH} = 1.00$
$a_{JET} = 0.50$
$a_{JET} = 0.60$
$a_{JET} = 0.40$

Total Error Rate: **0.94 %**

| car | face | airplane | motorbike |
| 0% | 0% | 1.00% | 2.75% |

DAS-DT(MFH-CH)

$a_{MFH} = 1.00$
$a_{CH} = 1.00$
$a_{CH} = 2.00$
$a_{CH} = 3.50$

Total Error Rate: **0.98 %**

| car | face | airplane | motorbike |
| 0% | 0.92% | 1.00% | 2.00% |

DAS-DT(MFH-CH-jet)

$a_{MFH} = 1.00$
$a_{CH} = 1.00$
$a_{JET} = 1.00$
$a_{CH} = 0.70$
$a_{JET} = 1.05$
$a_{CH} = 0.80$
$a_{JET} = 1.10$

Total Error Rate: **0.50 %**

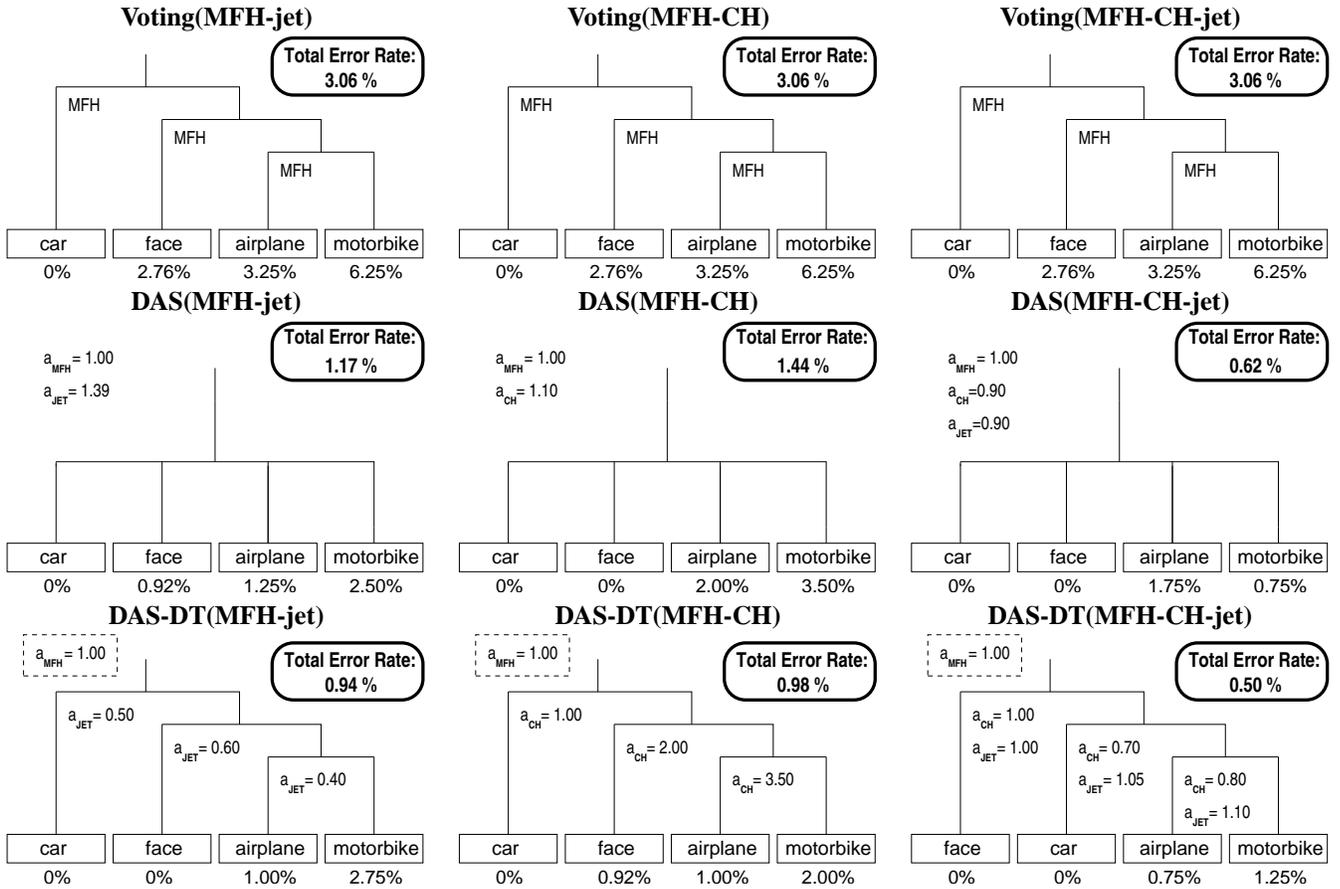| face | car | airplane | motorbike |
| 0% | 0% | 0.75% | 1.25% |

Table 5: Caltech database experiments: results for voting scheme (top row), DAS (middle row) and DAS-DT (bottom row). We report results for two-cues experiments (MFH-jet in the left column, and MFH-CH in the middle column) and the three-cues experiments (right column). Each graph reports the decision tree obtained, the total error rate and the error rate for category. We report also, for DAS and DAS-DT, the weighting coefficients found for each cue, at each leaf node. The MFH coefficient was found to be 1 for all these experiments; we report it on the left side of each graph. For the voting scheme, we give, at each leaf node, the cue chosen for the split.

With respect to the voting algorithm, we observe that, in general, the splitting between classes will be different using DAS-DT. This happens, for instance, when we use all the three cues (Table 5, right column, top and bottom row). We noticed the same behavior for the CogVis-ETH experiments (this result is not reported due to space constraints). From all these experiments, we can conclude that DAS, by itself or combined with a decision tree, is an effective method for cue integration and competitive with state of the art algorithms.

4 Conclusions

Robustness is a vital feature of any visual recognition system aiming to work in unconstrained, real-world settings. Cue integration is a fundamental strategy for achieving robustness. This paper presents a new algorithm for cue integration that extends the idea of accumulation to discriminative classifiers. The major strength of cue integration via accumulation is that, even when all cues are indicating as best hypothesis the wrong answer, the algorithm still has a chance to perform correctly. We illustrated this behavior by running experiments on several databases, using different types of features and comparing our results with a probabilistic accumulation scheme and a voting algorithm. Results confirm the effectiveness of our approach.

We plan to extend this work as follows: (1) we are currently extending our algorithm to increase the number and type of cues available. In some cases, this may mean using SVMs for the first time on certain type of features, and it may lead to the need of defining new kernel functions. (2) We plan to run extensive experiments on other visual pattern recognition applications, like material classification and ac-

tion recognition. We also plan to continue experiments on object recognition and categorization, testing robustness of our method with respect to scale, light changes and occlusions. (3) The impressive results obtained in [32] using Adaboost suggest that the extension of our algorithm in that direction could bring interesting results; we intend to explore this possibility. (4) We plan to use multi-class SVMs with the multi-cue Mercer kernel we derived, and compare results with those given by DAS. Also, the coefficients $\{a_p\}$ are obtained so far via cross validation, but a theoretical analysis of the generalization bound of DAS could lead to a more theoretically sound learning procedure.

Acknowledgemnts

References

[1] E. Allwein, R. Schapire, Y. Singer. Reducing multiclass to binary: a unifying approach for margin classifiers. *JMLR* 1, 2000.

[2] J. Aloimonos, D. Shulman. Integration of visual modules: an extension of the Marr paradigm. Academic Press, Boston, 1989.

[3] A. Barla, F. Odone, A. Verri. Hausdorff kernel for 3D object acquisition and detection. *ECCV02*.

[4] S. Belongie, C. Fowlkes, F. Chung, J. Malik. Partitioning with indefinite kernels using the Nystrom extension. *ECCV02*.

[5] H. Bulthoff, A. Yuille. Bayesian models for seeing shapes and depth. *Comments on Theoretical Biology*, 2(4), 1991.

[6] C. Brautigam, J-O Eklund, H. Christensen. A model-free approach for integrating multiple cues. *ECCV98*.

[7] B. Caputo, G. Dorko. How to combine color and shape information for object recognition: kernels do the trick. *NIPS02*.

[8] C. Chang, C. Lin. LIBSVM: a library for support vector machines. 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[9] J. Clark, A. Yuille. Data fusion for sensory information processing systems. Kluwer Academic Publisher, 1990.

[10] N. Cristianini, J. S. Taylor. An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press, 2000.

[11] R. Fergus, P. Perona, A. Zisserman. Object class recognition by unsupervised scale-invariant learning. *CVPR03*.

[12] G. L. Foresti, V. Murino, C. Ragazzoni, A. Trucco, A voting--based approach for fast object recognition in underwater acoustic images. *IEEE J. Ocean Eng*, 22(1), 1997.

[13] Y. Freund, R. Schapire. A decision-theoretic generalization of n-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1), 1997.

[14] D. R. Heisterkamp, J. Peng, H. K. Dai. Adaptive quasiconformal kernel metric for image retrieval. *CVPR01*.

[15] B. Leibe, B. Schiele. Analyzing appearance and contour based methods for object categorization. *CVPR03*.

[16] D. Lowe. Object recognition from local scale invariant features. *ICCV99*.

[17] J. Matas, R. Marik, J. Kittler. On representation and matching of multi-coloured objects. *ICCV95*.

[18] B. Mel. SEEMORE: combining color, shape and texture histogramming in a neurally-inspired approach to visual object recognition. *NC*, 9, 1997.

[19] H. Murase, S. Nayar. Visual learning and recognition of 3D objects from appearance. *IJCV*, 14, 1995.

[20] R. Nelson, A. Selinger. A cubist approach to object recognition. *ICCV98*.

[21] B. Parhami. Voting algorithms. *IEEE Trans on Reliability*, 43(3), 1994.

[22] T. Poggio, V. Torre, C. Koch. Computational vision and regularization theory. *Nature*, 317, 1985.

[23] S. Romdhani, P. Torr, B. Schölkopf, A. Blake. Computationally efficient face detector. *ICCV01*.

[24] D. Slater, G. Healey. Combining color and geometric information for the illumination invariant recognition of 3D objects. *ICCV95*.

[25] B. Schiele, J. L. Crowley. Recognition without correspondence using multidimensional receptive field histograms. *IJCV*, 1, 2000.

[26] Z. Sun. Adaptation for multiple cue integration. *CVPR03*.

[27] M. J. Swain, D. H. Ballard. Color Indexing. *IJCV*, 1991.

[28] C. Schmid, R. Mohr. Combining greyvalue invariants with local constraints for object recognition. *CVPR96*.

[29] D. Terzopoulos. Integrating visual information from multiple sources. From pixels to predicates, Ablex, Norwood, 1986.

[30] J. Triesch, C. Eckes. Object recognition with multiple feature types. *ICANN98*.

[31] V. Vapnik. Statistical learning theory. Wiley and Son, NY, 1998.

[32] P. Viola, M. Jones. Rapid object detection using a boosted cascade of simple features. *CVPR01*.

[33] W. Zhu, S. Wang, R. S. Lin, S. Levinson. Tracking of object with SVM regression. *CVPR01*.

[34] C. Wallraven, B. Caputo, A. Graf. Recognition with local features: the kernel recipe. *ICCV03*.

[35] L. Wolf, A. Shashua. Kernel principal angles for classification machines with application to image sequence interpretation. *CVPR03*.