World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:3, No:12, 2009

Integrating Context Priors into a Decision Tree Classification Scheme

Kasim Terzić, Bernd Neumann

Abstract—Scene interpretation systems need to match (often ambiguous) low-level input data to concepts from a high-level ontology. In many domains, these decisions are uncertain and benefit greatly from proper context. This paper demonstrates the use of decision trees for estimating class probabilities for regions described by feature vectors, and shows how context can be introduced in order to improve the matching performance.

Keywords—Classification, Decision Trees, Interpretation, Vision

I. Introduction

THERE is a growing interest in the field of computer vision for using high-level knowledge for interpreting scenes from a wide range of domains. This involves vision tasks which go beyond single-object detection to provide an explanation of the observed scene. These tasks include inferring missing and occluded parts and recognising structure and relationships between objects in the scene. Typical examples include monitoring tasks such as airport activity recognition [1], interpreting building façades [2]–[4] or analysing traffic situations [5], [6].

As shown in [7], scene interpretation can be formally modelled as a knowledge-based process. Such a knowledge-based system, based on the configuration methodology, exists in the form of SCENIC [8]. The SCENIC system consists of a domain-specific knowledge base of concepts and an interpretation process which propagates constraints, instantiates concepts to instances, determines relations between instances, etc. Concepts are mainly aggregate models, their instances represent aggregate instantiations (or simply: "aggregates"), i.e. configurations of concrete objects in scenes. The interpretation process attempts to recognise aggregates which describe the observed evidence.

The task of the middle layer of an interpretation system is then to match the detections from low-level image processing algorithms to concepts from the high-level domain ontology. There are many examples in the literature where specific classes of objects are detected in the image with high accuracy [9], [10]. However, many domains exist where the classes have heterogeneous appearances and where there is considerable overlap between appearances, leading to many classification errors when using a purely bottom-up approach. An example is the domain of building façades which consists mostly of rectangular objects of varying sizes and considerable overlap between classes (see Figures 2 and 3). Previous research [11], [12] has shown that a purely appearance-based classification is difficult in this domain, even when it is reduced to a 4-class problem (e.g. *Roof, Sky, Ground* and *Façade*) . In domains like these, it may be preferable to explicitly model the uncertainty of classification so that high-level context can improve the decision.

Department Informatik, Vogt-Kölln-Str. 30, 22527 Hamburg, Germany
{terzic|neumann}@informatik.uni-hamburg.de

Fig. 1. Example from the eTRIMS annotated façade database. Each object is marked by a bounding polygon and a label from the ontology (indicated here by different colours.)

This paper presents a multi-class classification scheme based on impure decision trees. A decision-tree classifier is automatically learnt for a given combination of classes and feature vectors, and its leaves carry the class probabilities for given evidence for all the classes in the ontology. In other words, it serves as a discrete approximation of the conditional probability density functions $P(C|E)$ for all the classes. As such, it can express the uncertainty of bottom-up classification and can be easily combined with contextual priors (e.g. coming from high-level interpretation) for disambiguation. While impure decision trees are well-known, we are not aware of their use for scene interpretation.

The approach is evaluated on a large database of annotated façade images. Three separate aspects of the decision trees are evaluated: bottom-up classification in the façade domain compared to SVMs (Section IV-B), the accuracy of probability estimates (Section IV-C), and the effect of using contextual priors on the classification rate (Section IV-D). In this paper, manually determined priors were used in order to measure the effect that correct context has on the classification rate. The integration of automatically calculated priors is planned in the future.

The following section introduces the domain and shows why it makes classification difficult. Section III, explains the classification methodology using decision trees. Section IV shows the evaluation of the performance on an annotated image database. Section V summarises our findings and discusses future work.

II. The Facade Domain

Recently, there has been an increased interest in interpreting building scenes, e.g. for localisation [13], vehicle navigation [6] and photogrammetry [14]. Buildings, being man-made structures, exhibit a lot of regularity that can be exploited by an interpretation system, but there is still enough variety within this structure to present a challenge for interpretation and learning tasks [15], [16].

A large database of annotated façade images exists as an outcome of the eTRIMS project [17], which can serve both as ground truth for classification and interpretation tasks, and as learning data. It contains close to 1000 fully annotated images. A sample image from the database is shown in Figure 1. The high-level ontology describing the domain used for the experiments in this paper contains the following classes: *Balcony, Building, Canopy, Car, Cornice, Chimney, Door, Dormer, Entrance, Façade, Gate, Ground, Pavement, Person, Railing, Road, Roof, Sign, Sky, Stairs, Vegetation, Wall, Window,* and *Window Array*. Some of these classes represent primitive objects without parts (like *Door* or *Window*), and some represent aggregates consisting of primitive objects (like *Balcony* or *Window Array*) or other aggregates (like *Façade*), thus forming a hierarchical structure.

For several reasons, the façade domain presents a number of challenges regarding classification:

- Most of the objects are rectangular (façades, windows, doors, railings, etc.) and of similar size. Some of the objects can come in virtually any colour (walls, doors, cars), some are semi-transparent (railings, vegetation) and blend with the objects behind them, and some can reflect other objects (windows and window panes). This leads to significant overlap between classes for most feature descriptors.
- The variability of appearance within each class is greater than the difference between classes, making it difficult to create compact appearance models.
- Some aggregate classes consist of parts in a loose configuration and as such don't have a characteristic appearance by themselves, e.g. balcony or façade.
- Some classes are distinctly more common than others. Around 55% of all annotated objects in the eTRIMS database are windows. Thus, a classifier seeking to minimise the expected total error will tend to misclassify objects as windows.

The difficulty of bottom-up classification in the façade domain was also discussed in [11] and [18]. On the other hand, the façade domain provides a lot of context which can be useful for classification. To name a few examples, entrances are usually located at the bottom of a façade, roofs at the top, windows are located in arrays, etc.

There are several approaches which exploit this context in the façade domain, using configuration [19], Markov Random Fields [16] or grammars [4] . Our approach uses a probabilistic model for context generation in terms of dynamic priors from a Bayesian Compositional Hierarchy (BCH) [20]. A BCH is a special kind of Bayesian Network with aggregates as nodes and isomorphic to the aggregate hierarchy. Dynamic priors are provided by propagating the effect of evidence assignments to other nodes of the BCH. Details of high-level interpretation, however, will not be provided in this paper, which focusses on the low-level stage using decision trees.

Fig. 2. Some windows from the annotated database.

Fig. 3. Some doors from the annotated database. The appearance and shape varies a lot and there is significant overlap with the Window class.

III. Learning Decision Trees

A decision tree is a tree where the leaf nodes represent classifications and each non-leaf node represents a decision rule acting on an attribute of the input sample. A sample described by a d-dimensional feature vector f and consisting of d scalar attributes is classified by evaluating the decision rule at the root node and passing the sample down to one of the subnodes depending on the result, until a leaf node is reached. The result is a partitioning of the feature space into labelled disjoint regions.

In a binary decision tree, the rules correspond to yes/no questions and each nonleaf node has exactly two children. The most common type of decision trees, called *univariate* decision trees, only act on one dimension at a time and thus result in an axis-parallel partitioning. The experiments described in this paper use univariate decision trees.

If the leaves are allowed to correspond to more than one class, they are called impure leaves. If each class in a leaf node is given a probability, such trees can be used to model the uncertainty of the classification result. Essentially, they provide a discrete approximation of a probability density function, where the discretisation can be irregular.

For many problems, decision trees have competitive performance compared to other classification schemes [21]. At the same time, they have the advantage of having a result that can be understood intuitively because they split the feature space into regions with axis-parallel boundaries. In addition to providing bottom-up classification of low-level evidence, they can also *describe* the visual appearance of high-level concepts by specifying a region of feature space. This is an appealing property for scene interpretation, because it simplifies top-down processing by making it possible to pass expectations of low-level appearance of expected objects to image processing algorithms in a compact and understandable form.

A. Learning algorithm

We now address some aspects of decision tree learning. Since the space of all possible trees is huge, greedy algorithms

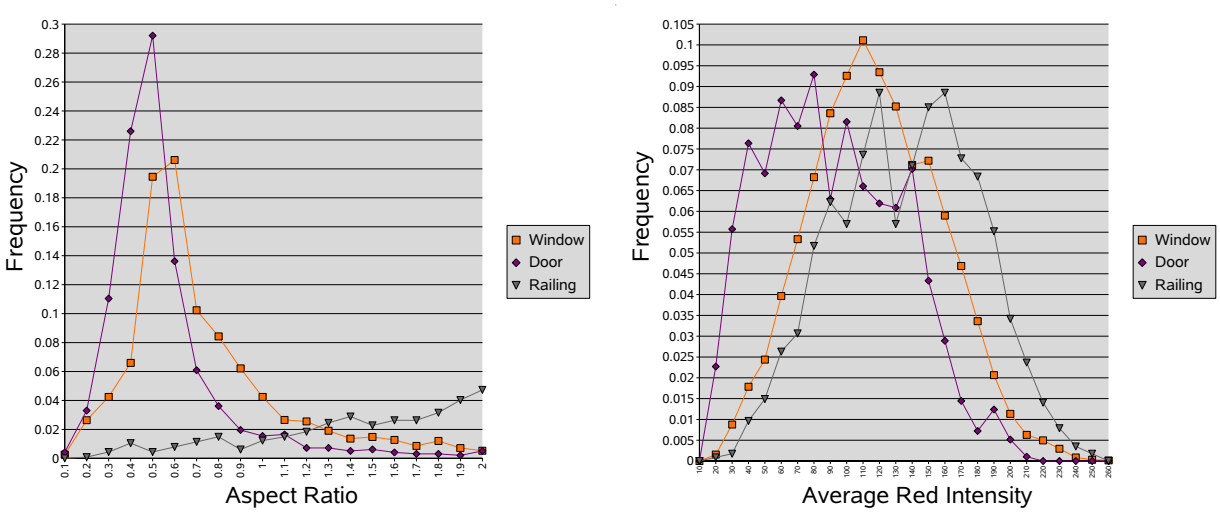


Fig. 4. Distribution of the aspect ratio (left) and average intensity of the red channel (right) for three common classes. It can be seen that making certain decisions based on these features is difficult even for the three-class case.

for learning the best tree are usually employed. The tree starts as a single root node containing all the samples and is recursively subdivided according to a splitting criterion. There are many splitting criteria in use for learning decision trees, two of the most popular are information content and the Gini coefficient[1]. For both criteria, the conditional probability $P(\omega|t)$ at a node t must be known, which can be approximated as

$$P(\omega_i|t) = \frac{N_i(t)}{N(t)}$$

where $N_i(t)$ is the number of samples in t belonging to class ω_i, and $N(t)$ is the number of all samples in t, where samples are taken from a training set.

The Gini coefficient is a measure of the impurity of a node, and as such also related to the information content of the node. The Gini coefficient of node t is defined as

$$G(t) = \sum_{i \neq j} P(\omega_i|t)P(\omega_j|t) \quad (1)$$

For a given split that divides t into t_l and t_r, the change in the Gini coefficient is given as

$$\Delta G(sp,t) = G(t) - (G(t_l)P_l + G(t_r)P_r)$$

where P_l and P_r are the priors for the left and right sub-node, respectively. The best split is the one that maximises $\Delta G(sp,t)$.

When learning a decision tree, the node with the highest impurity (measured as high information content or high Gini coefficient) is split in a way that maximises the splitting criterion. This entails two decisions: choosing the attribute (dimension) to split on and choosing its best value for the split. A simple approach, used for learning the trees described in this paper, is to perform an exhaustive search through the space of all possible splits in all possible dimensions, and to choose the one that minimises the Gini coefficient of the resulting sub-nodes. Given a set of n samples, each described by a d-dimensional feature vector f, an exhaustive search of all possible splits in each dimension has a complexity of $O(nd)$.

[1]More detailed explanations can be found in [22] and [21]

B. Pruning

Overfitting is a well-known problem with learning of decision trees [21], [22]. The leaf-splitting can be continued until all leaves have pure class membership. Such learnt trees describe the training set well, but if the data are not perfectly separable or contain noise, they do not generalise well to unseen examples, essentially modelling the noise in the training set. One can terminate the learning once a stopping criterion is fulfilled (e.g. minimum change of impurity function), or use one of many pruning algorithms to reduce the maximal tree.

Classification and Regression Trees (CART) were first introduced by Breiman [23] and still present a popular method for pruning learnt decision trees. The basic idea is to add a constant α to the impurity measure at each split, as a measure of the cost of additional complexity introduced by the split.

More specifically, if $R(t)$ is the measure of impurity at node t (the misclassification rate), then $R_\alpha(t) = R(t) + \alpha$ is the complexity measure of the node t. If $\tilde{T}$ is the set of all leaves in a tree T, and $|\tilde{T}|$ the cardinality of $\tilde{T}$, then $R(T) = \sum_{t \in \tilde{T}} R(t)$ is the estimated misclassification rate of a tree T, and

$$R_\alpha(T) = \sum_{t \in \tilde{T}} R_\alpha(t) = R(T) + \alpha|\tilde{T}|$$

is the estimated complexity-misclassification rate of T. If T_t is defined to be a subtree with node t at its root, the strength of the link from node t to its leaves can be calculated as

$$g(t) = \frac{R(t) - R(T_t)}{|\tilde{T}_t| - 1} \quad (2)$$

The nodes with a low $g(t)$ are punished as they add complexity without significantly improving the classification

result. The algorithm starts with the maximal tree and calculates $g(t)$ for all nodes. The node with the lowest value of $g(t)$ is made into a leaf, and all of its children are removed. The new values for $g(t)$ are calculated for all the predecessors of the affected node, and the process is repeated on the new tree.

The result is a succession of trees, starting with the initial, maximal tree, and ending with a tree containing only the root node. Each of these trees is a classifier. All the trees are tested on an unseen validation dataset and the tree with the best classification rate is selected as the final classifier.

C. Classification

If an impure leaf l contains samples of several classes, an estimate of $P(C|L)$ for all classes and leaves can be formulated as

$$P(c|l) = \frac{N_c(l)}{N(l)}$$

where $N_c(l)$ is the number of samples in l belonging to class c and $N(l)$ is the number of all samples in l. The probabilities at the leaves $P(C|L)$ reflect the success rate achieved with the training set used to learn the tree.

Instead of encoding $P(C|L)$, one can observe how often an object belonging to class c generates evidence described by the leaf l, giving the class-conditional probability $P(L|C)$ for all classes and leaves. Then, Bayes rule gives the posterior probability as

$$P(C|L) = \frac{P(L|C)P(C)}{P(L)} \tag{3}$$

Finding the class for which $P(C|L)$ is maximum gives a MAP classifier. Since each evidence sample e is mapped into a leaf l of the decision tree, $P(C|L)$ serves as a discrete approximation of $P(C|E)$.

D. Incorporating Context

The formulation in Equation 3 allows for introducing updated priors $P'(C)$, which reflect the scene context coming from incremental high-level intepretation or an additional knowledge source. If it is assumed that the typical appearance of the classes is not affected by context, i.e. that $P(L|C) = P'(L|C)$, the probability that a leaf l belongs to class c can be written as

$$P'(C|L) = \frac{P(L|C)P'(C)}{P'(L)} = \frac{P(L|C)P'(C)}{\sum_c P(L|C)P'(C)} \tag{4}$$

The leaves of a decision tree typically store $P(C|L)$ and not $P(L|C)$, so an update mechanism is derived to calculate $P'(C|L)$ from $P(C|L)$, $P(C)$ and $P'(C)$.

$$P'(C|L) = \frac{P'(CL)}{P'(L)} = \frac{P(L|C)P'(C)}{P'(L)} = \frac{P(L|C)P'(C)}{P'(L)}$$

$$= \frac{P(LC)\frac{P'(C)}{P(C)}}{P'(L)} = \frac{P(C|L)P(L)\frac{P'(C)}{P(C)}}{P'(L)} \tag{5}$$

$P'(L)$ can be expressed as

TABLE I
THE MEANS OF THE GAUSSIAN DISTRIBUTIONS USED TO GENERATE SYNTHETIC SAMPLES.

$\mu(class1)$	-0.5	-0.5	-0.5
$\mu(class2)$	0.5	0.5	0.5
$\mu(class3)$	-0.5	-0.5	0.5
$\mu(class4)$	0.5	0.5	-0.5
$\mu(class5)$	-0.5	0.5	-0.5
$\mu(class6)$	-0.5	0.5	0.5
$\mu(class7)$	0.5	-0.5	0.5
$\mu(class8)$	0.5	-0.5	-0.5

$$P'(L) = \sum_c P'(CL) = \sum_c P(L|C)P'(C)$$

$$= \sum_c P(LC)\frac{P'C)}{P(C)} = \sum_c P(C|L)P(L)\frac{P'(C)}{P(C)}$$

$$= P(L)\sum_c P(C|L)\frac{P'(C)}{P(C)} \tag{6}$$

Inserting 6 into 5 gives

$$P'(C|L) = \frac{P(C|L)\frac{P'(C)}{P(C)}}{\sum_c P(C|L)\frac{P'(C)}{P(C)}} \tag{7}$$

where $P(C)$ are the domain priors of the training set used for learning the tree, $P(C|L)$ are the conditional class probabilities at the leaves of the decision tree, and $P'(C)$ are the updated class priors.

IV. Evaluation

Our decision trees were tested on both synthetic data and annotated objects from the façade domain. In each case, the automatically learnt trees were tested as pure bottom-up classifiers, and then the effect of manually updated context priors on the classification rate was evaluated.

A. Synthetic Data

We first tested the decision trees on synthetic 4-class and 8-class data. Each sample is drawn from a 3-dimensional Gaussian distribution. Table I shows the means of the distributions, and Table II shows the standard deviations and the priors of the classes. The first dataset has four equally probable classes, the second set has classes chosen to be more similar to the façade domain, and the third set increases the standard deviation leading to more overlap between classes. Datasets 4 to 6 follow the same pattern, using 8 classes.

The results were compared with SVM-based multiclass classifiers using the svmlight software [24]. For each N-class dataset, three results were obtained: using the learnt decision tree followed by a MAP classification, by choosing the strongest response from N one-against-all SVM classifiers (referred to as *SVM1*), and finally by performing a majority vote among $N(N-1)/2$ pairwise SVM classifiers (referred to as *SVM2*). The 4-class datasets were tested using 10000 training samples (of which 1000 are used for pruning the

TABLE II
THE PRIORS ON THE CLASSES OF THE SYNTHETIC DATA AND THE STANDARD DEVIATION USED FOR THE GAUSSIAN DISTRIBUTIONS MODELLING THE CLASSES.

	DS 1	DS 2	DS 3	DS 4	DS 5	DS 6
P(class1)	0.25	0.1	0.1	0.125	0.05	0.05
P(class2)	0.25	0.1	0.1	0.125	0.05	0.05
P(class3)	0.25	0.2	0.2	0.125	0.05	0.05
P(class4)	0.25	0.6	0.6	0.125	0.05	0.05
P(class5)	0	0	0	0.125	0.05	0.05
P(class6)	0	0	0	0.125	0.10	0.10
P(class7)	0	0	0	0.125	0.10	0.10
P(class8)	0	0	0	0.125	0.55	0.55
σ_{1-8}	0.5	0.5	1.5	0.5	0.5	1.5

TABLE III
COMPARISON OF A ONE-AGAINST-ALL SVM CLASSIFIER (SVM1), A PAIRWISE SVM CLASSIFIER (SVM2), AND OUR DECISION TREE ON 6 SYNTHETIC DATASETS. THE NUMBERS REPRESENT THE CLASSIFICATION RATE FOR A GIVEN DATASET.

	SVM1	SVM2	Decision tree	
DS1	0.7805	0.7801	0.7654	(73 nodes)
DS2	0.8395	0.8412	0.8311	(109 nodes)
DS3	0.6843	0.6863	0.6745	(119 nodes)
DS4	0.5825	0.5915	0.5829	(411 nodes)
DS5	0.7238	0.7236	0.7145	(335 nodes)
DS6	0.5604	0.5793	0.5722	(25 nodes)

trees) and 10000 test samples. The 8-class datasets used 20000 training samples (2000 for pruning) and 20000 test samples.

The results, shown in Table III show the comparison of our approach with the SVM-based classifiers. The performance is within a percentage point of the SVM classifiers in almost all cases, outperforming one of the SVM classifiers on datasets 4 and 6.

B. Real Data

Our experiments on real data are based on the annotated façade image database from the eTRIMS project. All images are fully annotated using bounding polygons and class labels from a common ontology. For the experiments in this paper, 599 rectified images were used (façade edges are parallel to the image axes), consisting of 27922 objects in total. From these images, 15357 training objects, 6981 validation objects (used for pruning), and 5584 testing objects were used.

Table IV shows the composition of the 18-dimensional feature vector used to describe each object. It consists of simple and general features, because previous work on feature selection showed these features to be useful in the façade domain [11], [12], and more complex features such as statistical moments and colour histograms did not perform as well in our experiments.

The results of the decision tree classifier learnt for the 24-class façade object problem using the Gini coefficient for optimisation can be seen in Figure 5. The overall classification rate across all classes is 75.63%, with most classes showing a strong peak at the diagonal of the confusion matrix (see Figure 5).

It is apparent that the classes *Facade* and *Building* are often confused, as are *Road* and *Pavement*, but this is an expected

TABLE IV
THE COMPOSITION OF THE FEATURE VECTOR.

f_0	area
f_1	compactness: $4\pi \times area/perimeter^2$
f_2	aspect ratio: $width/height$
f_3	rectangularity: $area/(width \times height)$
f_{4-5}	mean and standard deviation of the red channel
f_{6-7}	mean and standard deviation of the blue channel
f_{8-9}	mean and standard deviation of the green channel
f_{10-18}	8-bin edge orientation histogram

TABLE V
COMPARISON OF A ONE-AGAINST-ALL SVM CLASSIFIER (SVM1), A PAIRWISE SVM CLASSIFIER (SVM2), AND OUR DECISION TREE ON 5584 OBJECTS FROM 599 ANNOTATED IMAGES FROM THE FAÇADE DOMAIN.

SVM1	SVM2	Decision tree	
0.7092	0.6999	0.7563	(601 nodes)

result, given how visually similar these classes often are. This is a point where high-level context (in terms of a prior expectation for the classes) could improve the classification results.

Another interesting result is the poor performance with classes *Sign, Chimney* and *Door*. In the case of *Sign* and *Chimney*, the prior of the classes is so low that classifying all of them as windows actually reduces the overall error rate. The prior of the class *Door* is quite high but, as shown in Figure 3, the visual appearance is often very close to the appearance of the *Window* class, which has a far higher prior. The solution to these problems is to introduce contextual information in the form of updated priors for different image regions. If there is a strong scene context suggesting one class over the other, this can be used for disambiguation, as will be shown in Section IV-D.

Once again, the results were compared with SVM-based multiclass classifiers using the svmlight software. We used two SVM-based classifiers: based on 24 one-against-all SVM classifiers (SVM1), and based on 276 pairwise SVM classifiers (SVM2). All three tests were performed on exactly the same objects, using the same features to keep results comparable. The only difference was that all individual features were scaled to between 0 and 1 for the SVMs. Since SVMs do not need a validation set, the objects used for pruning the decision tree were used as additional training objects for the SVMs. We used the default kernel (radial basis function) and default parameters (determined automatically by the svmlight software).

Table V shows the results. The confusion matrix for the one-against-all SVM classifier is shown in Figure 6. Our decision-tree based method outperformed both SVM-based methods in bottom-up classification. Another interesting observation is that the problems with classification of doors are even more pronounced when using SVM-based classifiers, as opposed to decision trees. The performance on the *Balcony* class is also worse.

C. Accuracy of probability estimates

One nice property of decision trees is that they provide an estimate of the probability of correct classification (without

	Balcony	Building	Canopy	Car	Chimney	Cornice	Door	Dormer	Entrance	Facade	Gate	Ground	Pavement	Person	Railing	Road	Roof	Sign	Sky	Stairs	Vegetation	Wall	Window	Window-Array
Balcony	106	2	0	1	0	0	0	0	5	2	0	0	1	0	5	0	16	0	0	0	7	0	77	9
Building	9	79	0	1	0	0	0	0	0	72	0	0	0	0	1	0	11	0	1	0	6	0	9	5
Canopy	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	3	0	0	1	0	2	1
Car	1	0	0	53	0	0	2	4	4	1	0	0	1	0	7	2	10	0	0	0	13	0	23	0
Chimney	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	13	0
Cornice	0	0	0	0	0	192	0	0	0	0	0	0	0	0	1	0	0	2	0	0	0	0	1	0
Door	0	0	0	0	0	0	49	0	1	0	0	0	0	0	0	0	0	0	0	0	6	0	149	2
Dormer	5	0	0	1	0	0	0	1	0	0	0	0	0	0	3	0	1	0	0	0	0	0	5	1
Entrance	3	0	0	1	0	0	4	0	14	4	0	0	0	0	2	0	3	0	0	0	2	0	40	4
Facade	8	43	0	0	0	0	1	1	0	145	0	0	0	0	1	0	5	0	0	0	3	0	10	6
Gate	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Ground	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	3	0	0	0	0	0	0	0
Pavement	4	0	0	4	0	0	0	0	0	1	0	0	15	0	9	11	10	0	0	1	8	0	11	2
Person	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Railing	9	0	1	4	0	3	0	1	0	0	0	0	2	0	137	1	5	2	0	0	7	0	51	9
Road	1	0	0	1	0	0	0	1	0	1	0	0	11	0	2	10	4	0	1	0	9	0	0	3
Roof	23	3	0	10	0	0	0	1	6	0	0	0	7	0	5	0	76	0	0	0	19	0	14	5
Sign	0	0	0	1	0	2	2	0	0	0	0	0	2	0	8	0	0	5	0	0	2	0	25	0
Sky	2	1	0	1	0	0	0	1	0	2	0	0	0	0	0	1	7	0	90	0	3	0	3	0
Stairs	2	0	0	2	0	0	0	1	0	0	0	0	0	0	3	0	1	0	0	1	1	0	6	0
Vegetation	6	8	0	7	0	0	0	0	4	2	0	0	7	0	7	2	21	0	1	0	102	0	22	3
Wall	0	1	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Window	34	3	0	3	0	1	39	0	2	0	0	0	1	0	28	0	5	1	1	1	8	0	2976	17
Window-Array	4	0	1	1	0	1	0	0	0	14	0	0	1	0	21	0	1	1	0	0	2	0	54	172

Fig. 5. Confusion matrix for the learnt decision tree. Overall classification rate is 75.63%.

TABLE VI
COMPARISON OF ESTIMATED CLASS PROBABILITIES FOR THE STRONGEST CLASS WITH THE ACTUAL CLASSIFICATION RATE. LEAVES WITH SIMILAR PROBABILITIES FOR THE STRONGEST CLASS WERE GROUPED TOGETHER IN BINS. THE LEFT COLUMN SHOWS THE EXPECTED RESULT (MEAN VALUE OF EACH BIN), AND THE OTHER COLUMNS SHOW THE ACTUALLY MEASURED CLASSIFICATION RATES FOR THESE LEAVES. THE BEST PROBABILITY ESTIMATES WERE OBSERVED WITH A SMOOTHED TREE AND M=1. IN ONE CASE, NO LEAVES HAD A PROBABILITY ESTIMATE IN THE GIVEN RANGE, THIS IS INDICATED AS "N/A".

Expected	No smoothing	m=0.1	m=0.5	**m=1**	m=5	m=10
0.95	0.92	0.94	0.94	**0.95**	0.95	0.95
0.85	0.81	0.84	0.85	**0.87**	0.89	n/a
0.75	0.65	0.67	0.74	**0.75**	0.82	0.88
0.65	0.49	0.50	0.62	**0.64**	0.72	0.71
0.55	0.45	0.44	0.50	**0.60**	0.67	0.54
0.45	0.36	0.36	0.43	**0.42**	0.45	0.27

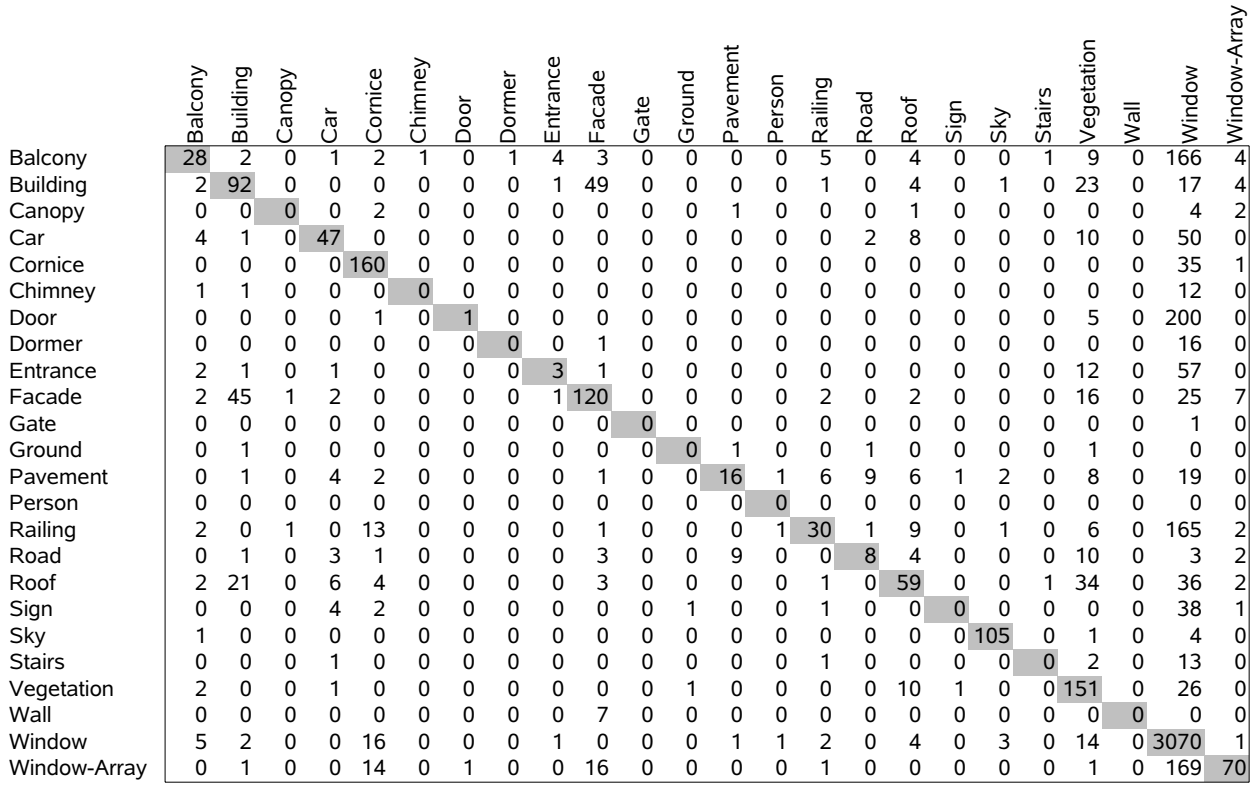
TABLE VII
COMPARISON OF CLASSIFICATION RATE FOR THE ORIGINAL TREE (LEFT COLUMN) AND TREES SMOOTHED WITH DIFFERENT VALUES OF M (RIGHT).

Original	m=0.1	m=0.5	m=1	m=5	m=10
0.7563	0.754835	0.752507	0.750895	0.708453	0.682307

consideration of context). In scene interpretation systems, this is useful information since it can be used to influence the order of interpretation steps. However, it is well-known that when trees are learnt in a way that tries to maximise the classification rate, the probability estimates are incorrect, especially for domains with unbalanced priors [25].

Several *probability smoothing* approaches have been proposed in the literature to address this problem [25]–[27]. A common and effective smoothing approach is m-estimation introduced by Cestnik [28]. The probability estimate at the leaves $P(c|l) = \frac{N_c(l)}{N(l)}$ is replaced by $P_s(c|l) = \frac{N_c(l)+P_d(c)m}{N(l)+m}$, where $P_d(c)$ is the domain prior for class c. We calculated the smoothed probabilities $P_s(C|L)$ for all classes and leaves. Since m-smoothing is a heuristic which affects different classes differently, all probabilities in all leaves were renormalised so they sum to one again. The parameter m determines how strongly the probabilities at the leaves are adjusted towards the domain prior. We determined the parameter m experimentally, as described below.

We compared the estimated probability provided by the leaves of the learnt decision trees with the actual classification rate. To this end, a comparison was made between a tree with no smoothing and a number of trees corresponding to different values for the parameter m. Ideally, the probability estimate of the decision tree will be the same as the probability observed in practice. In other words, if an object is classified as a window with $P(window|l) = 0.7$, it is expected that such a classification will be correct in 70% of the cases. In order



	Balcony	Building	Canopy	Car	Cornice	Chimney	Door	Dormer	Entrance	Facade	Gate	Ground	Pavement	Person	Railing	Road	Roof	Sign	Sky	Stairs	Vegetation	Wall	Window	Window-Array
Balcony	28	2	0	1	2	1	0	1	4	3	0	0	0	0	5	0	4	0	0	1	9	0	166	4
Building	2	92	0	0	0	0	0	0	1	49	0	0	0	0	1	0	4	0	1	0	23	0	17	4
Canopy	0	0	0	0	2	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	4	2
Car	4	1	0	47	0	0	0	0	0	0	0	0	0	0	0	2	8	0	0	0	10	0	50	0
Cornice	0	0	0	0	160	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	35	1
Chimney	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	12	0
Door	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	200	0
Dormer	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	16	0
Entrance	2	1	0	1	0	0	0	0	3	1	0	0	0	0	0	0	0	0	0	0	12	0	57	0
Facade	2	45	1	2	0	0	0	0	1	120	0	0	0	0	2	0	2	0	0	0	16	0	25	7
Gate	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Ground	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0
Pavement	0	1	0	4	2	0	0	0	0	1	0	0	16	1	6	9	6	1	2	0	8	0	19	0
Person	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Railing	2	0	1	0	13	0	0	0	0	1	0	0	0	1	30	1	9	0	1	0	6	0	165	2
Road	0	1	0	3	1	0	0	0	0	3	0	0	9	0	0	8	4	0	0	0	10	0	3	2
Roof	2	21	0	6	4	0	0	0	0	3	0	0	0	0	1	0	59	0	0	1	34	0	36	2
Sign	0	0	0	4	2	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	38	1
Sky	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	105	0	1	0	4	0
Stairs	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	2	0	13	0
Vegetation	2	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	10	1	0	0	151	0	26	0
Wall	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Window	5	2	0	0	16	0	0	0	1	0	0	0	1	1	2	0	4	0	3	0	14	0	3070	1
Window-Array	0	1	0	0	14	0	1	0	0	16	0	0	0	0	1	0	0	0	0	0	1	0	169	70

Fig. 6. Confusion matrix for the one-against-all SVM classifier (SVM1). Overall classification rate is 70.92%.

to test this, nodes with similar $P(c_{strongest}|l)$ were grouped together and the actual classification rate measured for each group.

Table VI summarises the results. We show the original tree (no smoothing) and smoothed trees using m = 0.1, 0.5, 1, 5 and 10. It can be seen that smoothing improves the probability estimates, and that the best results were achieved with m=1. One downside of smoothing is that it usually reduces classification accuracy. The effect of different smoothing factors on the classification rate is shown in Table VII. It can be seen that smoothing with m=1 doesn't impact classification rate strongly, and still significantly improves the probability estimates, making it the best choice for this domain.

D. Contextual information

We have tested the effect that changing the class priors has on the classification rate. We have simulated correct scene context by artificially altering the priors $P(C)$. For each tested object, the prior on the correct class was set to a certain value $P'(C)$ and all other priors renormalised so they sum all up to one again.

Figure 7 shows the effect on three different classes from the façade domain. The *Window* class has a very high domain prior (around 55%), the *Stairs* class has a very low domain prior (around 0.3%), and the *Door* class is relatively common (around 4%), but easily confused with the *Window* class. The graphs show that context is particularly helpful for less common and easily confused classes.

The context was simulated in these experiments, but it can be replaced by dynamic priors from Bayesian Compositional Hierarchies [20] or a similar probabilistic reasoning scheme in the future. The improvements shown in Figure 7 suggest that scene context in the form of updated priors will lead to improved classification.

V. Summary and Future Work

We have shown the application of decision trees to uncertain classification in a complex, multi-class domain. Decision trees offered competitive performance to standard multi-class SVM classification schemes on synthetic data, and better performance on the façade domain. At the same time, they allow easy incorporation of context in the form of class priors.

Currently, work is underway to integrate this middle-level classification framework into the scene interpretation system *SCENIC*. Also, the use of dynamic priors provided by a Bayes Compositional Hierarchy is being investigated. These context-specific priors on classes in a scene should improve the classification results, especially for visually similar and often-confused classes.

An interesting extension of this work is the feedback from high-level interpretation to image-processing algorithms. Decision trees offer a partitioning of the feature space into axis-

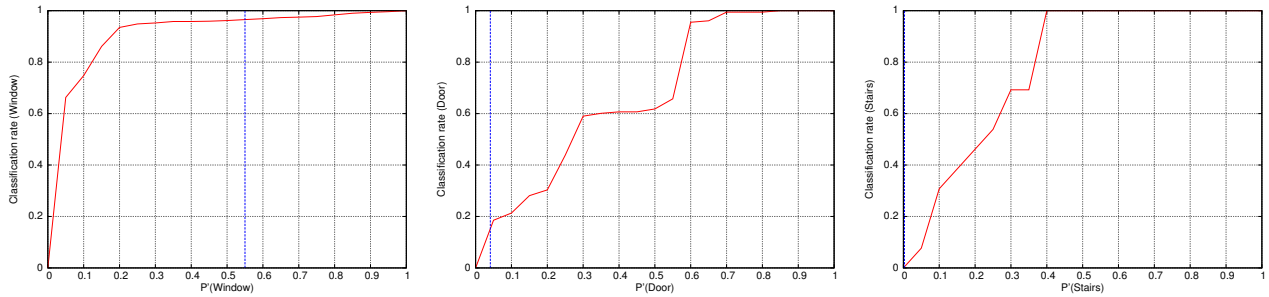


Fig. 7. The effect of updated prior $P'(C)$ on three classes from the façade domain. From left to right, they are: *Window, Door* and *Stairs*. The vertical blue line shows the classification using the domain prior $P(C)$ without any change. The curves for door and stairs are jagged because they were obtained from fewer samples, which were represented by fewer nodes.

parallel, easy-to-describe blocks. Given a strong expectation for a certain class of an object, it is possible to formulate a description of the object in terms of allowed feature ranges, which can help a low-level algorithm detect it.

ACKNOWLEDGEMENT

This research has been supported by the European Community under the grant IST 027113, eTRIMS - eTraining for Interpreting Images of Man-Made Scenes.

REFERENCES

[1] F. Fusier, V. Valentin, F. Bremond, M. Thonnat, M. Borg, D. Thirde, and J. Ferryman, "Video understanding for complex activity recognition," *Machine Vision and Applications (MVA)*, vol. 18, pp. 167–188, August 2007.

[2] L. Hotz, B. Neumann, and K. Terzić, "High-level expectations for low-level image processing," in *Proceedings of the 31st Annual German Conference on Artificial Intelligence*, Kaiserslautern, September 2008.

[3] S. Wenzel, M. Drauschke, and W. Förstner, "Detection of repeated structures in facade images," in *7th Open German / Russian Workshop on Pattern Recognition and Image Understanding*, E. Michaelsen, Ed. Ettlingen: FGAN-FOM, August 2007.

[4] J. Čech and R. Šára, "Language of the structural models for constrained image segmentation," Czech Technical University, Prague, Tech. Rep. Technical Report TN-eTRIMS-CMP-03-2007, 2007.

[5] M. Mohnhaupt and B. Neumann, "Understanding object motion: recognition, learning and spatiotemporal reasoning," pp. 65–91, 1993.

[6] B. Hummel, W. Thiemann, and I. Lulcheva, "Scene understanding of urban road intersections with description logic," in *Logic and Probability for Scene Interpretation*, ser. Dagstuhl Seminar Proceedings, A. G. Cohn, D. C. Hogg, R. Möller, and B. Neumann, Eds., no. 08091. Dagstuhl, Germany: Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, 2008. [Online]. Available: http://drops.dagstuhl.de/opus/volltexte/2008/1616

[7] L. Hotz and B. Neumann, "Scene interpretation as a configuration task," *KI*, vol. 19, no. 3, pp. 59–, 2005.

[8] K. Terzić, L. Hotz, and B. Neumann, "Division of work during behaviour recognition - the SCENIC approach," in *Workshop on Behaviour Modelling and Interpretation, 30th German Conference on Artificial Intelligence*, Osnabrück, Germany, September 2007.

[9] B. Leibe, E. Seemann, and B. Schiele, "Pedestrian detection in crowded scenes," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, 2005, pp. 878–885 vol. 1.

[10] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.

[11] M. Drauschke and W. Förstner, "Comparison of adaboost and adtboost for feature subset selection," in *PRIS 2008*, Barcelona, Spain, 2008.

[12] M. Drauschke and W. Förstner, "Selecting appropriate features for detecting buildings and building parts," in *21st Congress of the International Society for Photogrammetry and Remote Sensing (ISPRS)*, Beijing, China, 2008.

[13] U. Steinhoff, D. Omercevic, R. Perko, B. Schiele, and A. Leonardis, "How computer vision can help in outdoor positioning." in *AmI*, ser. Lecture Notes in Computer Science, B. Schiele, A. K. Dey, H. Gellersen, B. E. R. de Ruyter, M. Tscheligi, R. Wichert, E. H. L. Aarts, and A. P. Buchmann, Eds., vol. 4794. Springer, 2007, pp. 124–141.

[14] F. Korč and W. Förstner, "Interpreting terrestrial images of urban scenes using discriminative random fields," in *Proc. of the 21st Congress of the International Society for Photogrammetry and Remote Sensing (ISPRS)*, 2008.

[15] J. Hartz and B. Neumann, "Learning a knowledge base of ontological concepts for high-level scene interpretation," in *IEEE Proc. International Conference on Machine Learning and Applications*, Cincinnati (Ohio, USA), Dec 2007.

[16] D. Heesch and M. Petrou, "Markov random fields with asymmetric interactions for modelling spatial context in structured scenes," *Journal of Signal Processing Systems, to appear*, 2009.

[17] F. Korč and W. Förstner, "eTRIMS Image Database for interpreting images of man-made scenes," Tech. Rep. TR-IGG-P-2009-01, April 2009.

[18] V. A. Bochko and M. Petrou, "Recognition of structural parts of buildings using support vector machines," in *Pattern Recognition and Information Processing, PRIP2007*, 2007.

[19] L. Hotz, B. Neumann, K. Terzić, and J. Šochman, "Feedback between low-level and high-level image processing," Universität Hamburg, Hamburg, Tech. Rep. Report FBI-HH-B-278/07, 2007.

[20] B. Neumann, "Bayesian compositional hierarchies - a probabilistic structure for scene interpretation," Universität Hamburg, Department Informatik, Arbeitsbereich Kognitive Systeme, Tech. Rep. FBI-HH-B-282/08, May 2008.

[21] A. R. Webb, *Statistical Pattern Recognition, 2nd Edition*. John Wiley & Sons, October 2002.

[22] D. Poole, A. Mackworth, and R. Goebel, *Computational intelligence: a logical approach*. Oxford, UK: Oxford University Press, 1997.

[23] L. Breiman, J. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Monterey, CA: Wadsworth and Brooks, 1984.

[24] T. Joachims, "Making large-scale support vector machine learning practical," in *Advances in kernel methods: support vector learning*, B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds. Cambridge, MA, USA: MIT Press, 1999, pp. 169–184.

[25] B. Zadrozny and C. Elkan, "Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers," in *In Proceedings of the Eighteenth International Conference on Machine Learning*. Morgan Kaufmann, 2001, pp. 609–616.

[26] L. R. Bahl, P. F. Brown, P. V. De, and R. L. Mercer, "A tree-based statistical language model for natural language speech recognition," vol. 37, no. 7, Jul 1989, pp. 1001–1008.

[27] W. Buntine, "Learning classification trees," *Statistics and Computing*, vol. 2, pp. 63–73, 1992.

[28] B. Cestnik, "Estimating probabilities: A crucial task in machine learning," in *ECAI*, 1990, pp. 147–149.