A Probabilistic Approach to Image Orientation Detection via Confidence-based Integration of Low-Level and Semantic Cues

Jiebo Luo
Research and Development Laboratories
Eastman Kodak Company
jiebo.luo@kodak.com

Matthew Boutell
Department of Computer Science
University of Rochester
boutell@cs.rochester.edu

Abstract

Automatic image orientation detection for natural images is a useful, yet challenging research area. Humans use scene context and semantic object recognition to identify the correct image orientation. However, it is difficult for a computer to perform the task in the same way because current object recognition algorithms are extremely limited in their scope and robustness. As a result, existing orientation detection methods were built upon low-level vision features such as spatial distributions of color and texture. In addition, discrepant detection rates have been reported. We have developed a probabilistic approach to image orientation detection via confidence-based integration of low-level and semantic cues within a Bayesian framework. Our current accuracy is approaching 90% for unconstrained consumer photos, impressive given the findings of a psychophysical study conducted recently. The proposed framework is an attempt to bridge the gap between computer and human vision systems, and is applicable to other problems involving semantic scene content understanding.

1. Introduction

With an explosion in the popularity of both online and offline consumer image collections, organizing and accessing these images become challenging tasks. Digital or scanned images in the database are required to be displayed in their correct orientations. Unfortunately, this is currently done manually and automating the process can save time and labor. Furthermore, many image processing and vision algorithms (e.g., region-based approaches to content-based image retrieval) assume *a priori* knowledge of the image orientation. Again, automatic orientation is desirable.

Perception of image orientation is interesting. The orientation of some classes of images is clear and seems easy to detect; for instance, landscape images tend to contain sky on the top of the image and land on the bottom. At the other end of the spectrum, some images, e.g., close-ups of a plain rug, have no clear orientation, and some images have an orientation only discernible to a human through subtle context cues.

Automatically determining the orientation of an arbitrary image is a problem that attracted attention only recently. Existing systems use low-level features (e.g., color, texture) and statistical pattern recognition techniques. Such systems are exemplar-based, relying on learning patterns from a training set [1, 2] and without direct reference to semantic content of the images. Vailaya *et al.* originally reported 95+% accuracies on an image set derived from the Corel database [1]. More recently, Wang and Zhang reported a much lower accuracy of 78% on a different subset of Corel images using a similar, yet more sophisticated, method [2].

The discrepancies in accuracies cannot be explained other than that the databases were different. Current scene classification systems such as [1] enjoy limited success on *constrained* image sets such as Corel. However, with consumer images, the typical consumer pays less attention to composition and lighting than would a professional photographer, causing the captured scene to look less prototypical and, thus, not match any of the training exemplars well. The greater variability of consumer images, both in terms of color and composition, causes the high performance on clean, professional stock photo libraries of many existing systems to decline remarkably because it is difficult for exemplar-based systems to account for such variation in their training sets.

Major differences exist between Corel stock photos and typical consumer photos [3], including but not limited to: (1) Corel images used in [1] are predominantly outdoor and frequently with sky present, while there are roughly equal numbers of indoor and outdoor consumer pictures; (2) over 70% of consumer photos contain people, while it is the opposite with Corel; (3) Corel photos of people are usually portraits or pictures of a crowd, while in consumer photos the typical subject distance is 4-10 feet (thus containing visible, yet not dominating, faces); and (4) consumer photos usually contain a much higher level of background clutter. These differences have a profound impact on the algorithm performance and demand a more rigorous approach.

A rigorous psychophysical study was conducted recently to investigate the perception of image orientation in [4]. A collection of 1000 images (a mix of professional photos and consumer snapshots) was used in this study. Each image was examined by at least five observers and shown at varying resolutions. At each resolution, observers were asked to indicate the image orientation, the level of confidence, and the cues they used to make the decision. This study suggests that for typical images, the accuracy is close to 98% when using *all* available semantic cues recognizable by humans from high-resolution images, and 84% if only low-level vision features and coarse semantics observable (by humans) from thumbnails are used. The accuracies by human observers provide upper bounds for the performance of an automatic system. In addition, the use of a large, carefully chosen image set that spans the photo space (in terms of occasions and subject matter) and extensive interaction with the human observers revealed cues used by humans at various image resolutions: sky and people are the most useful and reliable among a number of important semantic cues.

Given the findings of the human observer study and the difficulty of current object detection algorithms on unconstrained images, we believe automatic detection of image orientation is still largely an unsolved problem.

We also strongly believe that semantic cues can bridge the so-called "semantic gap" between computer and human vision systems. When available, they can be incorporated to significantly improve the performance of an image understanding system.

2. Probabilistic Cue Integration Framework

We believe a viable approach to semantic image understanding has to address the following issues:

- ❑ Need to account for the viability and limitations of low-level feature-based approach
- ❑ Need to integrate semantic features when available
- ❑ Need to integrate features of different nature and frequency of occurrence
- ❑ Need to integrate critical domain knowledge
- ❑ Need to have good generalizability because of limited ground-truth training data

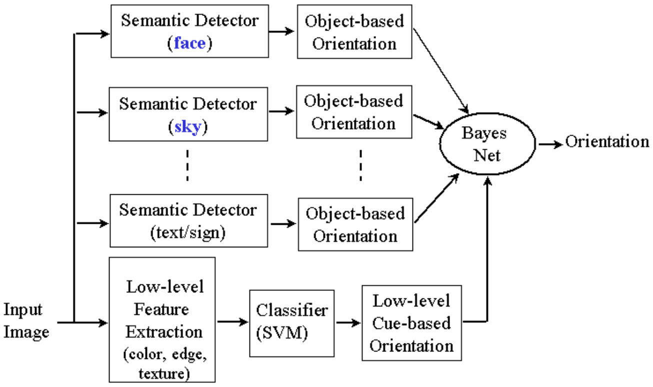


Figure 1: An integrated approach.

Figure 1 illustrates the proposed general framework for semantic understanding of natural images as a solution to these issues. The input is a digital image of a natural scene. Two sets of descriptors are extracted from the image: the first set corresponds to low-level features, such as color, texture, and edges; the second set corresponds to semantic objects that can be automatically detected. The low-level features can be extracted on a pixel or block basis, using a bank of pre-determined filters aimed at extracting color, texture or edge characteristics from the image. The semantic features are obtained using a bank of pre-designed object-based predictors that have reasonable accuracy at predicting image orientation (e.g., at least better than chance). The state of the art in object detection, both in terms of accuracy and speed, limits what is included in the object detector bank. The hybrid streams of low-level and semantic evidences are piped into a Bayesian network-based inference engine. The Bayes net is capable of incorporating domain knowledge as well as dealing with a variable number of input evidences, and produces semantic predicates.

3. Learning by Example Using Low-Level Cues

The goal of automatic image orientation detection is to classify an arbitrary image into one of four compass directions (N, S, E, W), depending on which direction the top of the image is facing. Doing so, based on low-level color and texture features alone, is a difficult problem. We designed a baseline system using low-level features and a one-vs-all SVM classifier, which is similar to, and achieved similar results to that in [2]. We made several improvements to boost computational efficiency and generalizability, and to suit the overall probabilistic inference scheme.

3.1. Feature Extraction

In our implementation of spatial color moments, we transform the image into LUV color space, divide it into 49 blocks using a 7 x 7 grid and compute the mean and variance of each band. Using this coarser grid gave similar accuracy and greater efficiency than the finer grids reported in [1, 2]. The 49 x 2 x 3 = 294 features correspond, in essence, to a low-resolution version of the image and crude texture features. One should not expect a classifier based on this feature set to match human performance when high-resolution images are available for viewing.

Edge direction can also give cues to the orientation of the image, especially in urban scenes. We follow the treatment in [2] and calculate a spatial edge direction histogram on the luminance band of the image as follows: divide the image into a 5 x 5 bin and extract edges using a Canny detector. For each block, we quantize the edge direction into 16 bins (22.5 degree increments) and add a bin for the percentage of non-edge pixels present in the block. This gives 17 x 25 = 425 features (vs 925 in [2]). Using a smaller number of bins helps generalizability (in consumer images) and increases efficiency.

3.2 Pruning the Training Set

Based on an understanding of the low-level inference engine, we identified certain types of images that cannot be accurately classified using low-level color moments or edge directions. These images would either confuse the SVM training, because of the wide variety of positions in which the colors would occur, or become support vectors that add no value for generalization. This is also supported by the findings of the human observer study in [4].

The following types are pruned from the training set (and *not* the test set): homogeneous textures, close-up views (e.g., flowers, people, animals), ambiguous orientations (e.g., aerial views), underwater images, reflections in the water, overly cluttered images (e.g., outdoor market scenes with no sky in image), indoor scenes with confusing lighting/busy ceilings, images where semantic cues are expected to work well while low-level cues are not, and so on. Examples of such pruned training samples are shown in Figure 2.

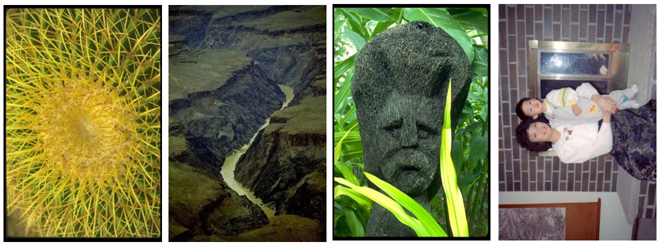


Figure 2: Examples pruned from the training set.

One advantage of pruning the training set is that the number of support vectors (and, thus, classification time) decreases dramatically. It also increases generalizability of the low-level classifiers.

3.3 Deriving Confidence for Probability Integration

Low-level features, such as color moments or edge direction histograms, give scene orientation cues. We have used an SVM classifier within a one-vs-all framework [5] to determine the orientation from these features. Within this framework, the SVM generates four real-value outputs for each image, corresponding to the image being rotated into four potential orientations. The image is classified with the orientation yielding the maximum output.

In anticipation of a Bayesian network that operates on *confidence* (for probabilistic integration of cues), we have discretized the output, into *strong* vs *weak* evidence because inference is much more efficient on discrete data (unless the data is assumed to be normally distributed).

In the one-vs-all framework, two measures have been used to determine rejection thresholds [2], and are good candidates for determining the strength of the SVM signal. First is the magnitude of the *maximum* output of the four. For example, if the maximum is negative (i.e., all four outputs are negative), the signal is extremely weak. Second is the *difference* between the top two SVM scores. If the difference is small, there is conflicting evidence in the image features, causing the SVM to classify the image with multiple orientations. Intuitively, if the maximum score is large and positive, and the difference between it and the next highest output is also large, then the output is unambiguous. We would like to call these outputs "strong" and other outputs "weak."

Our goal, therefore, is to use these two measures to determine a decision boundary between strong and weak SVM evidence. First, consider the distribution of *maximum* vs. *difference* for the SVM color moment output on a representative set of images (Figure 3). Points marked with blue triangles are correctly classified, while those marked with red circles are incorrectly classified. Because the data seems to be spread in a long, thin cloud, we transform it using PCA; the decision surface is chosen perpendicular to the direction of greatest variance [6] and such that 90% of the data on the "strong" side is classified correctly. This technique is repeated on the edge direction histogram feature set in a similar manner.

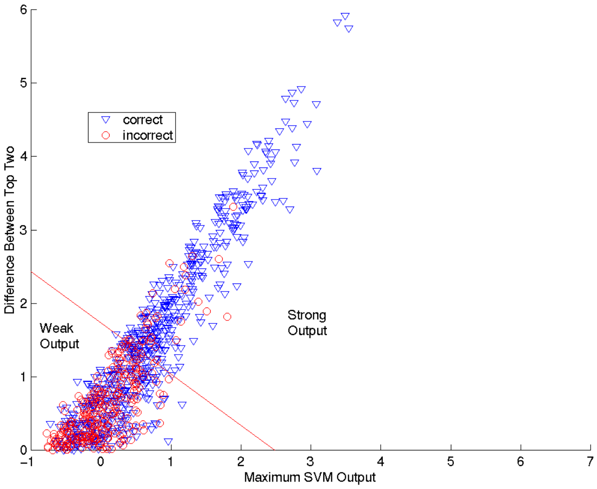


Figure 3: Maximum SVM score vs difference between the top two scores for color moment features. The decision boundary yields 90% classification accuracy on the points to the upper-right of it.

4. Inference by Semantic Cues

Semantic cues are selected based on their correlation to image orientation, occurrence, and confidence of the corresponding detectors we can build. We chose to use the following cues: face, blue sky, cloudy sky, white ceiling/wall, and grass, in an order of decreasing usefulness, which is supported by the psychophysical study [4]. Other semantic cues, such as open water, building, furniture, cars, flowers, and text, incur diminishing returns and increasing difficulties for building the associated detector. For example, text seems useful but its low occurrence and the language variety one needs to handle makes it unattractive.

Appropriate inference schemes need to be designed according to the nature of the selected cues and the robustness of the corresponding detectors. The detectors are described in a summary fashion in the following subsections with particular focus on the related orientation inference algorithms.

4.1. Orientation by Face Detection

We detect human faces using an algorithm based on Schneiderman's [7]. In addition, necessary improvements have been incorporated to make the original algorithm more robust for unconstrained, consumer type of images. Using a set of heuristic rules, the faces detected at all four possible orientations are combined to make an overall orientation decision. Note that it does not need quadruple the time to process an image for inferring orientation because the significant overhead of algorithm (e.g., feature extraction and model initialization) is only incurred once.

It is important for each detected face to be associated with a confidence level in order to derive the optimal decision on the image orientation. The output of the face detector is continuous, akin to a probability with higher scores indicating stronger confidence. From the distribution of *P*(score | face) and *P*(score | nonface) obtained using an independent validation set of 600 images, we were able to determine optimal thresholds, T_{strong} and T_{weak}, for declaring STRONG, WEAK, or NO face detection. We label orientation using the following pseudo-code:

```
if $score_{max} >= T_{strong}$
    label frame with orientation that produces $score_{max}$
    with STRONG confidence
elseif $score_{max} >= T_{weak}$
        if a single orientation has more faces detected
                label frame as that orientation
                with WEAK confidence
        else no orientation labeling
else no orientation labeling
```

Faces have proven to be a very strong cue for finding the correct image orientation, based on the strength of the face detector and the strong correlation between the face orientation and the image orientation (only 1% exception). For portrait images, strong face-based orientation classifications were 99% correct. For consumer images, strong classifications were 90% correct while declining to label 55% of images; including weak classifications led to 81% accuracy with 42% of images unlabeled.

4.2. Orientation by Blue Sky Detection

Sky is one of the most important subject matters frequently seen in photographs. It has been recognized that sky provides a strong cue to natural image understanding. Color has been the central feature of existing work on sky detection. Ironically, in order to increase sky detection accuracy, many researchers had to assume the image is an outdoor scene and its orientation is known [8]. It is probably the reason why sky detection has not been used for image orientation to date (i.e., "chicken and egg" problem).

In this study, we adopted a physical model-based blue sky detector as reported in [9] because it can infer the sky orientation by itself. As a result of the physics of light scattering by small particles in the air, clear sky often appears in the shade of deep, saturated blue at the top of the image and gradually desaturates to almost white towards a distant horizon line in the image. The gradient in the sky, rather than the location of sky, gives away the orientation of the image. Furthermore, the detector is unlikely to be fooled by other similarly colored objects, such as bodies of water, walls, toys, and clothes. These two advantages are vital for using blue sky to infer image orientation.

The blue sky detection algorithm in [9] detects large clear blue sky regions in two stages. In the first stage, a multilayer neural network performs pixel classification based on color and texture features. The RGB values of each pixel are used directly as the color features (the trained neural network performs the proper color transformation). The output of the pixel classification is a map of continuous "probability" values (not binary yet). Next, an image-dependent adaptive threshold is selected to obtain candidate sky regions after connected component analysis. In the second stage, a sky signature validation algorithm is used to eliminate false positive regions. First, the orientation of sky is inferred from the vertical/horizontal gradients in each extracted region. Finally, the algorithm determines a probability to reflect how well the region fits the model [9].

Using this algorithm for orientation is straightforward, except we need to account for the detection confidence levels as well as multiple detected sky regions. The same procedure was used to determine the optimal thresholds, T_{strong} and T_{weak}, for declaring STRONG, WEAK, or NO blue sky detection.

```
if $score_{max} >= T_{strong}$
    if an orientation has more strong blue sky regions
        label frame as that orientation
        with STRONG confidence
    else
        label frame with orientation that produces the
        largest product of $p = score \times \sqrt{area}$
        with STRONG confidence
elseif $score_{max} >= T_{weak}$
    if an orientation has more blue sky regions detected
        label frame as that orientation
        with WEAK confidence
    else no orientation labeling
else no orientation labeling
```

Blue sky detection turns out to be even more reliable (96% accuracy with no exception in its correlation to image orientation) than face detection for determining image orientation, and there is no need to analyze all four possible orientations because the detector itself can infer orientation. The only limitation is that blue sky does not appear as often as faces in consumer photos (only 22% of the time).

One potential issue with using sky detection is that sky orientation may have been somewhat captured implicitly by statistical learning of the low-level feature-based method. We have investigated this issue using a commercial grade Bayesian network analysis package based on the methodology by Cooper and Herskovits [11]. This package found no correlation between the *orientation predictions* by the specific sky detection algorithm used in this study and by the low-level color moment-based classifier. While correlation does not necessarily indicate causality, which is the basis for building Bayes networks, the lack of correlation affirms the absence of causality. Intuitively, it also made sense as the orientation is predicted using sky gradient, which is independent of sky location (which is what the low-level feature-based methods actually learned).

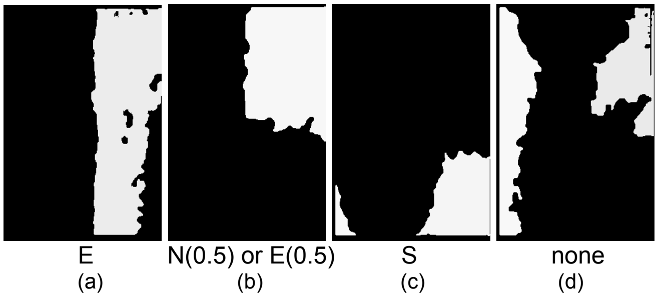


Figure 4: Orientation determined by the spatial configuration of detected **cloudy sky** regions. (a) and (b) are clear. The only possible explanation for (c) is *south* (bottom-up), while in (d) the opposite locations of the two regions and the fact that one region extends all the way along one border leads to "no decision."

4.3. Orientation by Cloudy Sky Detection

Unlike clear blue skies, cloudy/overcast skies have less unique color characteristics (i.e., the desaturation effect). Also, there are a large number of other semantic object classes, such as roads, walls, clothing, and snow that have very similar color and texture characteristics to cloudy/overcast skies. Thus, we have to build a different model for detecting cloudy/overcast sky regions. We use a combination of color and texture features to extract candidate cloudy/overcast sky regions in an image. These are further analyzed to eliminate the false positives.

A neural network, similar to that used for detecting clear blue skies, is used for performing the pixel-level classification. We have observed that sky regions (blue or cloudy/overcast) tend to be the brightest regions in an image because sky is almost always the main source of illumination in outdoor scenes. Converting the image to the LUV color space allows us to take advantage of this observation. A normalization step is used to define a normalized luminance feature l' on a per image basis, i.e., $l' = l / l_{\max}$, where $l_{\max}$ is the maximum raw luminance over the entire image. This physics-motivated feature, though less rigorous, leads to significant reduction in false positive detection of other grayish colored subject matter as cloudy/overcast sky. The normalized LUV triplet for each pixel provides the three color features for the neural network classifier. Six texture features are computed using a wavelet transform based on multiresolution analysis.

While image orientation can be directly inferred from face and blue sky, only the spatial configuration of cloudy sky region(s) provides indication of the most plausible image orientation. Heuristics need to be designed carefully to handle typical spatial configurations as shown in Figure 4.

Unlike with faces and blue sky, the image orientation cannot be inferred directly from the detected cloudy regions; the *spatial configurations* hold the key to image orientation. This is the case with ceiling/wall and grass as well. Furthermore, this and the remainder of semantic cues are weaker in nature because sometimes they point to a few possible orientations (as opposed to a single orientation). For example, from the cloud region in Figure 4(b), it is equally likely that the true orientation is east or north. In the remainder of this section, the pseudo-code for the heuristics, though far from trivial, is omitted due to space constraints.

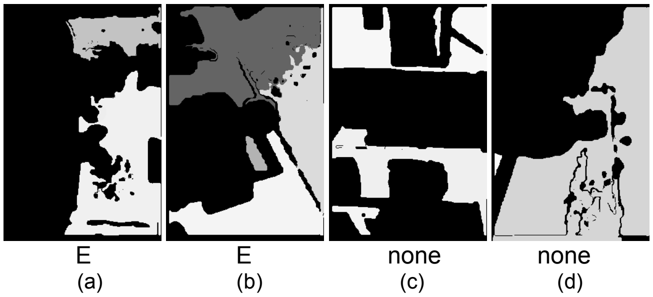


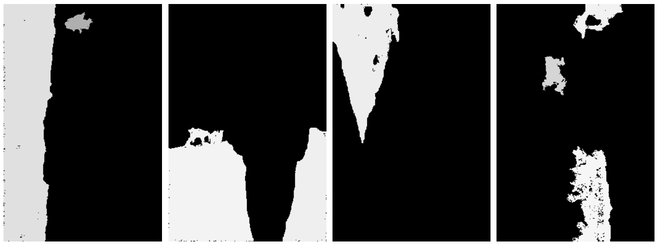
Figure 5: Orientation determined by the spatial configuration of detected white **ceiling/wall** regions. (a) is simple (east is the top), but (b) is tricky; three sides filling up makes *east* (right side) the only plausible top. It is clear that no decision can be made for (c) while careful consideration leads to also a "no decision" in (d).

4.4. Orientation by White Ceiling/Wall Detection

The white ceiling/wall detector is essentially the same as the cloudy sky detector because of the similar color and texture characteristics. The main difference is that a higher degree of occlusion is expected to occur with wall and ceiling (often connected), and more straight lines may be present. The heuristic rules for inferring orientation are somewhat different; e.g., if three neighboring borders are filled up with the fourth border partially open, the opposite border is the top. Examples are shown in Figure 5.

4.5. Orientation by Grass Detection

The grass detector is designed and trained similarly to the cloudy sky detector, except the luminance feature is not normalized. In addition, green foliage regions (trees) are used as negative examples in order to differentiate grass from trees because tree foliage tends to be in the top of the image while grass tends to be at the bottom. Furthermore, it is likely to see grass in the middle of an image, leaving the two directions perpendicular to the expanse of the grass area as the possible tops of the image. Heuristic rules are designed accordingly. A few typical spatial configurations of (potential) grass regions are shown in Figure 6.



E (a) N (b) E(0.7) or S(0.3) (c) E(0.5) or W(0.5) (d)

Figure 6: Orientation determined by the spatial configuration of detected **grass** regions. (a) and (b) are clear. (c) points most likely to the *east* (right side) and also likely *south* (bottom side). (d) points to either *east* or *west* (left or right sides).

5. Confidence-based Cue Integration

The various predictors may not classify an image as the same orientation. How does one arbitrate when the classifications by the predictors differ? Duin [10] discussed two types of combiners, fixed and trained. Fixed combining rules include voting and using the average [2] of the scores.

In this study, we chose to use a combiner in the form of a trained Bayes network (BN). Because the predictors differ in their nature, occurrence, and confidence, a statistically optimal way is to combine them in the probability domain (vs. a monolithic feature vector). A Bayesian classifier according to the *maximum a posteriori* (MAP) criterion gives orientation $\omega \in \{N,E,S,W\}$ by:

$$\hat{\omega} = \arg\max P(\hat{\omega} \mid S, L) = \arg\max P(S \mid \omega) P(L \mid \omega) P(\omega)$$

where S=semantic cues, L=low-level cues, and P(ω) = prior.

Determining the structure of the BN is straightforward once various *conditional* independencies between various cues are factored based on domain knowledge. The BN is shown in Figure 7. Note that we deliberately separate the actual detectors so that any improvement in a detector can be readily incorporated without re-training the network; only the associated detector's confusion matrix needs to be replaced at the bottom level of the network.

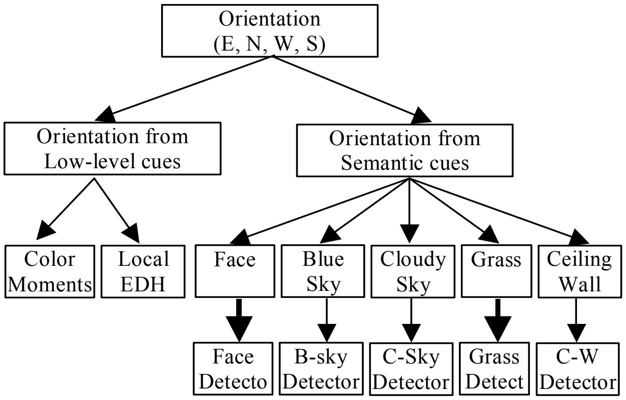


Figure 7: Structure of the Bayes network.

The parameters of the Bayes network, i.e., the individual conditional probability matrices (CPMs), were obtained from training. Two examples of the CPMs are included below. In the CPM headings, FES means "Face East Strong Detection", FEW means "Face East Weak Detection", ND = "No Detection", etc. Note that ND occurs with all semantic cues. The first example is related to face, which is a strong cue, and to the highlighted edge in Figure 7. The other example is related to grass, which is a weak cue, and to the other highlighted edge. It is noteworthy that the different types of features are weighted naturally according to their statistical significance and confidence. In comparison, ad hoc weighting schemes would be needed if the cues were to be integrated in a monolithic feature vector.

FES	FEW	FNS	FNW	FWS	FWW	FSS	FSW	ND	
0.60	0.20	0.01	0.02	0.01	0.02	0.01	0.02	0.11	FaceEast
0.01	0.02	0.60	0.20	0.01	0.02	0.01	0.02	0.11	FaceNorth
0.01	0.02	0.01	0.02	0.60	0.20	0.01	0.02	0.11	FaceWest
0.01	0.02	0.01	0.02	0.01	0.02	0.60	0.20	0.11	FaceSouth
0.01	0.02	0.01	0.02	0.01	0.02	0.01	0.02	0.88	NoFace

GE	GN	GW	GW	ND	
0.60	0.08	0.16	0.08	0.08	GrassEast
0.05	0.60	0.05	0.22	0.08	GrassNorth
0.16	0.08	0.60	0.08	0.08	GrassWest
0.05	0.22	0.05	0.60	0.08	GrassSouth
0.02	0.02	0.02	0.02	0.92	NoGrass

Another important advantage of the proposed probabilistic approach is that the prior of orientations can be readily incorporated at the root node of the Bayes network. For consumer photos (film scans), extensive studies showed that the priors of the four orientations are roughly [3]:

East 0.72 North 0.14 West 0.12 South 0.02

6. Experimental Results

We conducted extensive experiments to illustrate the performance of the proposed algorithm under various configurations. We trained on a mixture of 4572 (1136 per class) Corel images and 3744 (936 per class) consumer images. Our independent testing set consists exclusively of 3652 (913 per class) consumer images. We tested the accuracy of the low-level-based classifiers on a representative set of Corel images and obtained accuracy similar to [2]. The accuracies (using MAP estimation) on the consumer set are given in Table 1 and show the incremental effect of adding semantic cues. We conducted separate experiments using equal priors and the actual priors of the consumer images. Image orientations were manually changed in the equal-prior experiments, while the original orientations were used in the actual-prior experiments. The last row on the table represents thresholding the final integrated belief values to decide strong/weak confidence.

The minimum goal for the automatic algorithm is to beat the prior, which is, as previously stated, 72% in the landscape orientation ("east"). Note that east is the default landscape orientation in this data set. While daunting to beat, the effect of such a heavily skewed prior is apparent: the incorporation of the prior boosts the accuracy by 12% when only low-level cues ("CM+EDH") are used.

Table 1: **Incremental** accuracy on consumer images.

Type of Cues Used	Priors	
	True **(.72 .14 .12, .02)**	Equal **(.25 .25 .25 .25)**
CM only	*	68.8%
EDH only	*	54.7%
CM + EDH	82.7%	70.4%
CM + EDH + face	88.0%	77.8%
CM + EDH + face + sky	89.0%	81.2%
Add grass, ceiling, and cloudy sky	**89.7%**	**82.7%**
Using belief output of the BN as a reject option	91.3% on best 96%, 52.6% on other 4%	88.3% on best 88%, 43.3% on other 12%

* When classifying individual color and edge features using SVM, it is impossible to incorporate class priors.

Face detection added significant gains in accuracy: 7.4% with equal prior and 5.3% with the true prior. Note that these are additional gains. The 3.4% gain from adding blue sky is significant, given that the "easy" cases (e.g., those in Figure 8) had already been classified correctly by the low-level predictors. This gain represents those images with sky regions either too small to influence the low-level predictor (e.g., Figure 9a) or at an unusual location (e.g., Figure 9d). Note that the effect of blue sky is less pronounced (1%) with the true prior because blue sky tends to appear more often in landscape images.

The remaining, weaker semantic cues added approximately another 1% in accuracy, more so (1.5%) in the equal prior case. Again, the potential gain is mitigated by lower occurrences, weaker correlation or higher uncertainty (one or more sides still possible), lower confidence of the cue detectors (particularly more misleading false positives), and the fact that clear, "easy" cases with these cues present have already been claimed by the low-level predictors.

A "reject" option was introduced in [2] so that higher accuracy can be obtained while leaving some images unclassified. One interesting finding of the study in [4] is that there is extremely high correlation between the accuracy of human predictions and the associated human confidence. In our probabilistic scheme, the rejection option is particularly natural: simply threshold the final belief values of the Bayes network. Note that this is even more useful in the equal prior case (Table 1).

The examples in Figures 8 and 9 demonstrate the effects of the proposed system. Figure 8 contains examples where the low-level predictor is extremely effective and the reason is obvious. In these cases, additional semantic cues (e.g., sky) add no real value.

The benefit of the semantic cues is illustrated in Figure 9 where semantic cues either override the incorrect classification by the low-level predictors (a, b, d, e) or strengthen correct but perhaps weak predictions (f, h). In particular, the low level cues predicted "south" for (a) and (b) because the tops of these images are darker and more textured; "east" for (d) because the sky region is predominately on that side; and "north" for (e) because the person wearing darker clothes is at the bottom. Strong faces (b, e, h), strong blue sky (a, d), and strong cloudy sky (f) were the winning factors, while grass helped limit the possible orientations to two out of four orientations (f, h).

Meanwhile, low-level cues prove valuable when semantic cues are absent or not detected by the automatic algorithms; no sky was detected for the sunset scene (c) and no face was detected for the small and turned heads in (g).

We also analyzed the failure cases of the integrated system. Some typical examples are shown in Figure 10. These failures were due to concept failure (a: face of a person lying down), false positive detection (b, c: faces), and no semantic cues when low-level cues are incorrect (d).

On a SunBlade-1000 workstation, the algorithm takes about 6 sec./image: 5 for low-level feature extraction and classification (un-optimized) and the remaining for semantic feature extraction (optimized) and orientation prediction.

7. Discussions and Conclusions

This work represents an attempt to mimic the human approach to an intrinsically semantic image understanding problem such as image orientation. We believe this is a *general* approach as long as reasonably reliable semantic vision cues are selected according to the domain of the problem to supplement a reasonably reliable baseline of low-level vision features-based inference engine built using the principle of learning-by-example. The confidence levels associated with the cues, both low-level and semantic, play a critical role in reaching a final decision when cues agree and disagree. The key to successful application of this scheme to an image-understanding problem is domain knowledge, which guides the selection of the cues and construction of the probabilistic network for cue integration.

One alternative scheme is a cascade of predictors starting from the strongest one, akin to a decision tree [6]. However, this only works if all predictors are fairly strong and the strongest one has 90%+ accuracy, otherwise the overall accuracy will be below 90%. While our face and blue sky detectors are robust *and* image orientation can be robustly inferred from them, the other predictors are weak.

In conclusion, we developed an effective approach to image orientation detection by integrating low-level and semantic features within a probabilistic framework. Using all the available cues, our current accuracy is approaching 90% for unconstrained consumer photos; without taking advantage of the prior, the accuracy is approaching the average of humans when viewing thumbnails. The proposed framework is a successful attempt at bridging the gap between computer and human vision systems and is applicable to other scene understanding problems.

Acknowledgment: We would like to thank Henry Nicponski for his help with orientation from face detection.

References

[1] A. Vailaya, H. J. Zhang, and A. Jain, "Automatic image orientation detection," *Proceedings of IEEE International Conference on Image Processing,* 1999.

[2] Y. Wang and H. Zhang, "Content-based image orientation detection with support vector machines," *Proceedings of IEEE Workshop on Content-Based Access of Image and Video Libraries*, 2001.

[3] R. Segur, "Using photographic space to improve the evaluation of consumer cameras," *Proceedings of IS&T PICS Conference*, 2000.

[4] J. Luo, D. Crandall, A. Singhal, M. Boutell, and R.T. Gray, "Psychophysical study of image orientation perception," *Spatial Vision*, vol. 16, no. 5, 2003.

[5] B. Scholkopf, C. Burges, and A. Smola, *Advances in Kernel Methods: Support Vector Learning*, MIT Press, Cambridge, MA, 1999.

[6] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, John Wiley & Sons, New York, NY, 2001.

[7] H. Schneiderman, *A statistical approach to 3D object detection applied to faces and cars*, PhD thesis, CMU-RI-TR-00-06, Carnegie Mellon University, 2000.

[8] A. Vailaya and A. Jain, "Detecting sky and vegetation in outdoor images," *Proceedings of SPIE: Storage and Retrieval for Image and Video Databases VIII*, vol. 3972, 2000.

[9] J. Luo and S. P. Etz, "A physical model-based approach to sky detection in photographic images," *IEEE Transactions on Image Processing*, vol. 11, no. 3, pp. 201-212, 2002.

[10] R. P .W. Duin, "The combining classifier: To train or not to train?" Proceedings of International Conference on Pattern Recognition, 2002.

[11] G. Cooper and E. Herskovits, "A Bayesian method for the induction of probabilistic networks from data," *Machine Learning*, vol. 9, pp. 309-347, 1992.

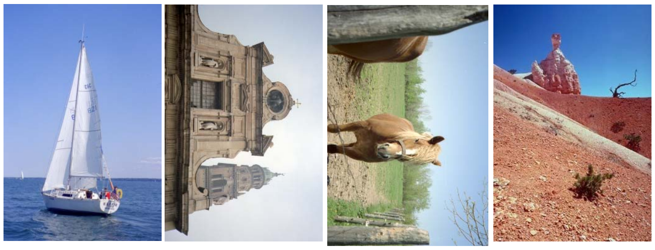


Figure 8: Examples for which the low-level feature–based approach is effective.

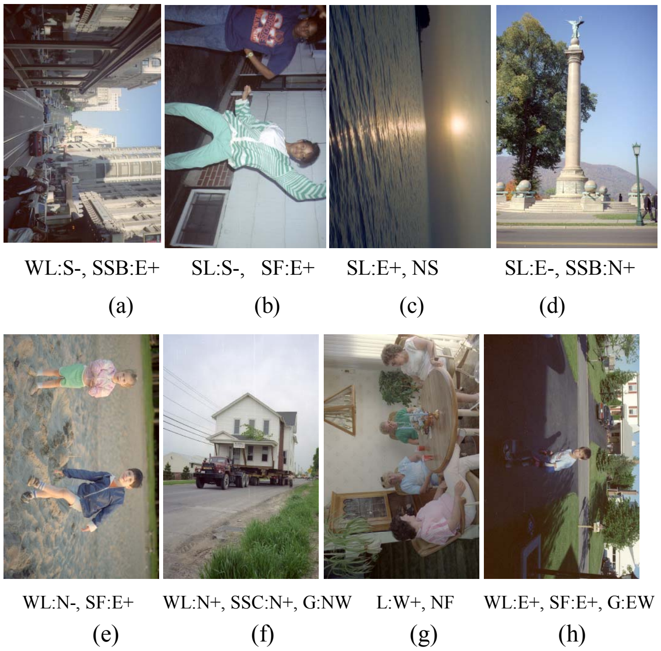


Figure 9: Examples for which the integrated approach is successful. The images are shown in their original orientation. The notations are: 'E/N/W/S' for the four orientations of *east*, *north*, *west* and *south*, '+' for correct prediction, '-' for incorrect prediction, 'W' for weak prediction, 'S' for strong prediction, 'L' for low-level cues, 'F' for face, 'SB' for blue sky, 'SC' for cloudy sky, 'G' for grass, 'NS' for no sky detection, and 'NF' for no face detection.

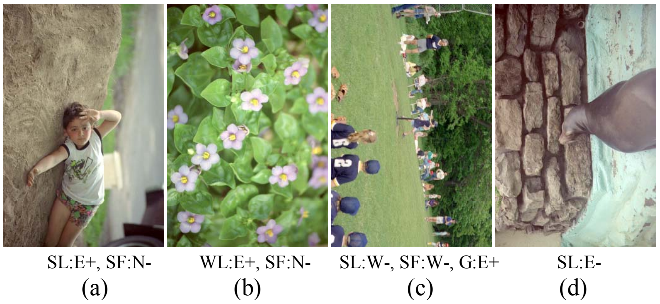


Figure 10: Examples of the failures of the integrated approach. These failures were due to concept failure (a: face of a person lying down), false positive detection (b, c: faces), and no semantic cues when low-level cues are incorrect (d).