Automatic Image Orientation Detection via Confidence-Based Integration of Low-Level and Semantic Cues

Jiebo Luo, *Senior Member*, *IEEE*, and Matthew Boutell, *Student Member*, *IEEE Computer Society*

Abstract—Automatic image orientation detection for natural images is a useful, yet challenging research topic. 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. Discrepant detection rates have been reported for these methods in the literature. 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 90 percent 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.

Index Terms—Image orientation, semantic cues, low-level cues, Bayesian networks, probabilistic inference, classification confidence.

1 Introduction

1.1 Problem Statement

With an explosion in the popularity of both online and offline consumer image collections, organizing and accessing these images become challenging tasks. Whether stored in personal albums, e-mailed, or posted to the Web, digital or scanned images in image libraries and personal albums are required to be displayed in their correct orientations (Fig. 1). Unfortunately, this is currently done manually and automating the process can save time and labor. Furthermore, many image understanding and image processing algorithms (e.g., content-based image retrieval systems) 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 (Fig. 2). In the remainder of this paper, a compass notation of (*north, east, west, south*) will be used for convenient differentiation of the four likely image orientations.

1.2 Recent Research

Automatically determining the orientation of an arbitrary image is a problem that only recently attracted attention in the research community, mostly as a result of the boom in digital image libraries. While some content-based image retrieval (CBIR) systems use features invariant to rotation (e.g., color histograms), many others assume that all the images in the library are in their upright orientation; these include those dependent upon object features and their spatial layouts (e.g., composite region templates [16]). All the current systems for determining image orientation 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 the semantic content of the images. Vailaya et al. originally reported 98 percent accuracies on an image set derived from the Corel database [1]. More recently, Wang and Zhang reported a much lower accuracy of 78 percent on a different subset of Corel images using a similar, yet more sophisticated, method [2]: Both color moment and edge direction features were classified using a Support Vector Machine extended to four classes using a one-versus-all approach. Our low-level cue-based classifier described in Section 3 is similar in principle to that used in [2].

The discrepancy in accuracies is most likely due to the fact that the databases were different. Current scene classification systems, such as [1], reported success on *constrained* image sets, such as Corel. Furthermore, 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

- *J. Luo is with the Research and Development Laboratories, Eastman Kodak Company, Rochester, NY 14650. E-mail: jiebo.luo@kodak.com.*
- *M. Boutell is with the Department of Computer Science, University of Rochester, Rochester, NY 14627. E-mail: boutell@cs.rochester.edu.*

Manuscript received 21 Oct. 2003; revised 15 Apr. 2004; accepted 24 Sept. 2004; published online 11 Mar. 2005.
Recommended for acceptance by R. Chellappa.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0329-1003.

0162-8828/05/$20.00 © 2005 IEEE Published by the IEEE Computer Society

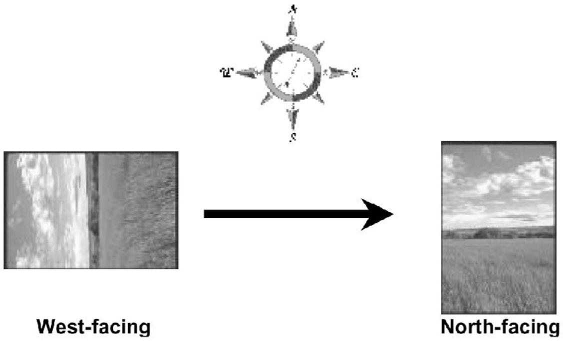


Fig. 1. The image orientation detection problem.

1. roughly equal numbers of indoor and outdoor consumer pictures;
2. more than 70 percent 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);
4. consumer photos usually contain a much higher level of background clutter;
5. consumer photos vary more in composition because 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; and
6. the structure of the Corel library (100 related images per CD) often leads to some correlation between training and testing sets.

These differences almost always cause the high performance on clean, professional stock photo libraries of many existing systems to decline markedly because it is difficult for exemplar-based systems to account for such variation in their training sets; analysis of consumer snapshots demands a more sophisticated approach.

1.3 A Psychophysical Study on the Perception of Orientation

A rigorous psychophysical study was conducted recently to investigate the perception of image orientation [4]. A collection of 1,000 images (a mix of professional photos and consumer snapshots, intended to span reasonably well the "photo space" [3] in terms of, e.g., picture seasons, occasions, locations, indoor/outdoor, people/no people) 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. Examples of these images are shown in [4].

This study suggests that, for typical images, human accuracy is close to 98 percent when viewing high-resolution (512×768) images, but declines to about 84 percent when the resolution of the images is reduced to 64×96. At the full resolution, all available semantic cues recognizable by humans are available and the subjects stated that they used primarily semantic cues at this resolution. However, low-resolution images force the subjects to use only low-level vision features and "coarse" semantics they could observe (or guess) from the thumbnails. Intuitively, the accuracies by human observers provide upper bounds for the performance of an automatic system because humans are trained to perform well in this recognition task (through subconscious daily exercise of holding books, newspapers, pictures, and even 3D objects in the proper orientation). Humans also possess remarkable power for object recognition in natural images (relative to state-of-the-art artificial intelligence systems).

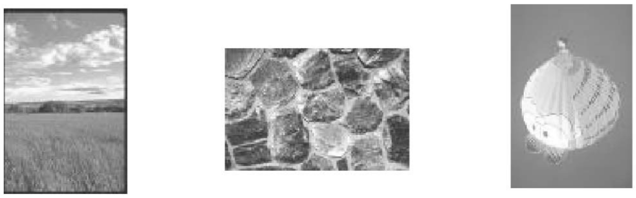


Fig. 2. The difficulty in image orientation detection: While some images follow clear patterns discernable using low-level features (e.g., color for landscape images), others require higher-level reasoning to discern (the hot-air balloon).

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: Semantic cues are very important for human interpretation of image orientation. In our study, only 1.6 percent of images were oriented correctly without mentioning any semantic cues. Some cues stood out as being very important. Sky was used in 31 percent of correct observations and people were used in 37 percent of correct observations. In fact, *sky*, *grass*, and *people* are the most useful and reliable among a number of important semantic cues, accounting for more than *70 percent* of the correct observations. Other cues include *clouds, water, trees, animals* (all species), *buildings* (all styles), *ground* (pavement, sand, dirt, carpet, floor, etc.), *furniture* (all types), and *vehicles* (all types).

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

We believe that a small but powerful set of computable semantic cues can help bridge the so-called "semantic gap" between computer and human vision systems. When available, they can be incorporated to improve the performance of an image understanding system, such as image orientation detection, as the psychovisual study suggests heavy reliance on semantics by humans.

This paper is organized as follows: We present our probabilistic framework for integrating low-level and semantic cues in Section 2. Image orientation using low-level cues is described in Section 3. Section 4 describes the semantic cues and the process of inferring orientation from these cues. We discuss the details of the Bayesian network used for cue combination in Section 5. Section 6 presents

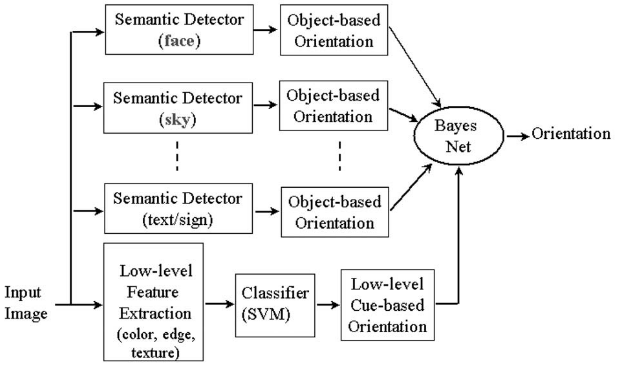


Fig. 3. An integrated approach to image orientation detection using low-level and semantic cues.

our experimental results on a set of consumer photographs. We conclude in Section 7.

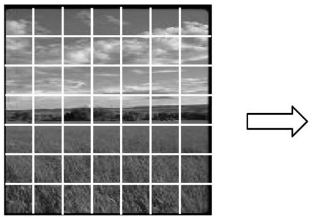
2 A Probabilistic Framework for Integrating Low-Level and Semantic Cues

Image orientation detection is an image understanding problem. We believe a viable approach to image understanding needs to address the following issues:

- Account for both the strength and weakness of a low-level feature-based approach.
- Integrate semantic features when available.
- Integrate features of different nature and frequency of occurrence.
- Integrate critical domain knowledge.
- Have good generalizability because of limited ground-truth training data.

Fig. 3 illustrates the general framework for orientation detection of natural images proposed as a solution to these issues. The input is a digital image of a photograph. 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 predetermined filters aimed at extracting color, texture, or edge characteristics from the image. The semantic features are obtained using a bank of predesigned 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, which serves as an arbitrator for various evidences that may or may not agree with each other. The Bayesian network is capable of incorporating domain knowledge as well as dealing with a variable number of input evidences, and produces semantic predicates.

We now discuss the specific low-level and semantic features, how to extract evidence for image orientation from these features, and how to integrate them using the Bayesian network.



$$x = \left.\begin{pmatrix} 0.4561 \\ 0.1928 \\ \ldots \\ 0.2756 \end{pmatrix}\right\} 294\,\text{dim}.$$

Fig. 4. Color moment (CM) features. See text for details.

3 Learning by Example Using Low-Level Cues

Determining the orientation of an arbitrary image 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-versus-all SVM (Support Vector Machine) classifier [5], which is similar to and achieved similar accuracy to that in [2]. However, we made several improvements to boost computational efficiency and generalizability and to suit the overall probabilistic inference scheme. First, we developed a more realistic training set consisting of both professional (Corel) and consumer photographs. Second, we pruned the training set of outliers (e.g., images with ambiguous or subtle orientation cues); this decreases the classifier complexity while boosting generalizability. Third, rather than rejecting test images with low confidence values during the classification, we retain them, along with the associated confidence values, and defer the final decision until all the cues are integrated. For more information on SVMs and other suitable classifiers, please see [5], [6], [14].

3.1 Feature Extraction: Color Moments (CM) and Edge Direction Histogram (EDH)

In our implementation of spatial color moments, we transform the image into the LUV color space, divide it into 49 blocks using a 7×7 grid, and compute the first two moments (mean and variance) of each of the three color bands. Color spaces such as LUV, in which color is decorrelated into luminance and chrominance components, have been used commonly and shown to be effective for image segmentation [17], [18] and orientation detection [1], [2]. Using this coarser grid, shown in Fig. 4, gave similar accuracy and greater efficiency than the finer grids reported in [1], [2]. The 294 $(49 \times 2 \times 3)$ features are normalized and correspond, in essence, to a low-resolution version of the image and crude texture features. One should not expect a classifier based on these features to match human performance (when viewing high-resolution images).

Edge direction histograms (EDH) 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 5×5 grid 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 nonedge pixels present in the block. This gives $17 \times 25 = 425$ features (versus 925 in [2]). Using fewer bins helps generalizability (in consumer images) and increases efficiency. Fig. 5 illustrates this

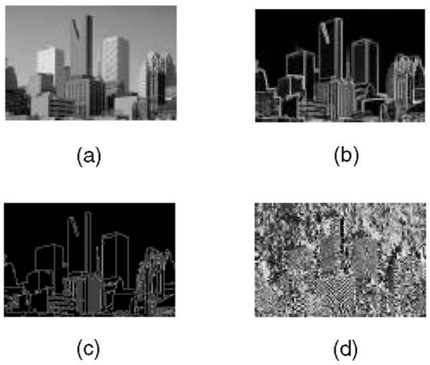


Fig. 5. Edge direction histogram (EDH) features. (a) Original. (b) Edge Magnitude. (c) Edge Map. (d) Edge Angles. In the angle image (d), edge directions are represented by gray levels. The final edge direction histogram is computed from (d) only over the mask of edge pixels shown in (c).

process; the final EDH is computed from Fig. 5d only over the mask of edge pixels shown in Fig. 5c.

It is helpful to have a good understanding of the performance by the color moment-based orientation classifier. Using color moments alone, we were only able to report a success rate of about 74 percent (on a Corel test set), compared to the 98 percent accuracy reported in [1]. This was after extensive communications with the primary author of the paper regarding the algorithm details. We believe such a discrepancy in performance by essentially the same algorithm can only be explained by significant differences in the databases. It was stated in [1] that the algorithm was "near perfect" on "long distance, outdoor scenes, and images with sky." Let us assume *"near perfect"*= 99.9 percent and denote the set of such "easy" images S1 and the set of remaining "challenging" images *S2*. Assuming one can manage to achieve an accuracy of 70 percent on set S2, which is nontrivial for a low-level feature-based method, then it follows that 93 percent of the total images have to be in set S1 in order to reach the reported overall accuracy of 98 percent. In other words, a predominant percentage of the images used in [1] should have been "easy" images, which is unlikely to be the scenario in a practical application. Incidentally, our accuracy using color moments alone is very close to what was reported later in [2] using a different (and perhaps more realistic) set of images. In summary, without the access to the specific data sets, we believe that the difference in data sets, as opposed to the methodology or implementation of the algorithms, is the reason for the differences in reported accuracy. The training and testing images used by our algorithm came from both Corel and our consumer photo databases. Examples of the consumer images can be found in the appendix.

Edge direction is a complementary, but less reliable (55 percent accuracy on consumer images, 67 percent accuracy on Corel images), predictor of image orientation than color moments. In the examples shown in Fig. 6, EDH correctly predicted the orientation for the first two images for which the color moments failed. However, EDH is not effective when there are no strong oriented edges in the image or when there are conflicting dominant edge orientations in the image, as shown by the last two images where both strong horizontal and vertical edges are present.

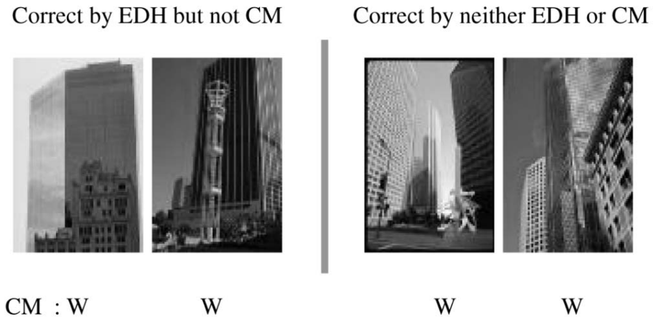


Fig. 6. Examples of images demonstrating the effectiveness of color moment (CM) and edge-direction histogram (EDH) low-level features.

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 result in support vectors covering outliers that add no value for generalization. Note that many of the pruned images have an unambiguous correct orientation; they were removed because the low-level features used here are simply incapable of sensing the subtle cues used by humans.

The following types are pruned from the training set (but *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), indoor scenes with confusing lighting/busy ceilings, and images where semantic cues are expected to work well while low-level cues are not. Examples of such pruned training samples are shown in Fig. 7. Overall, approximately 23 percent of the images in the training set were pruned, resulting in a 35 percent reduction in the number of support vectors.

One advantage of pruning the training set is that the number of support vectors (and, thus, classification time) decreases dramatically while keeping the classification accuracy relatively constant. It also increases generalizability of the low-level classifiers. Specifically, performance on the Corel set decreases only by 0.6 percent, while that on the consumer set actually increases by 0.9 percent.

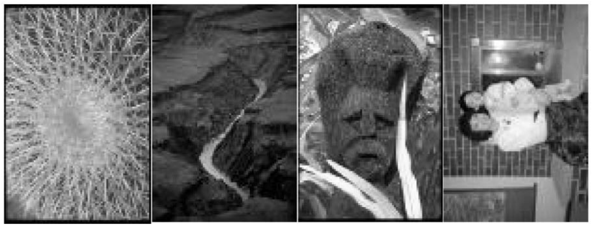


Fig. 7. Examples pruned from the training set because they are outliers (e.g., have ambiguous orientations or add little value to the generalizability of the classifier).

3.3 Deriving Confidence for Probability Integration

Low-level features, such as color moments or edge direction histograms, give scene orientation cues. We use an SVM classifier within a one-versus-all framework [5], [13] 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.

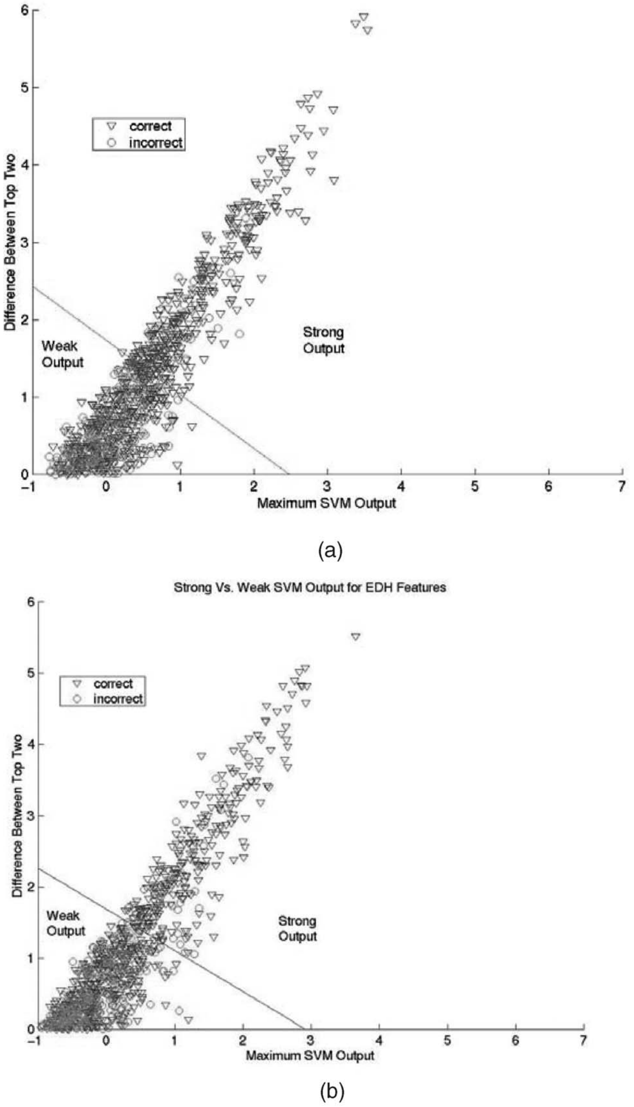
In anticipation of a Bayesian network that operates on *confidence* (for probabilistic integration of cues), we have discretized the output into *strong* versus *weak* evidence because inference is much more efficient on discrete data (unless the data is assumed to be normally distributed).

In the one-versus-all framework, two measures have been used to determine rejection thresholds [2] and, thus, 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 versus difference for the SVM color moment output on a representative set of images (Fig. 8a). Points marked with triangles are correctly classified, while those marked with circles are incorrectly classified. Because the data seems to be spread in a long, thin cloud, we transform it using Principal Component Analysis (PCA); the decision surface is chosen perpendicular to the direction of greatest variance [6]. Our intention is to introduce a soft decision by defining a "weak zone" where the distributions of true positives and false positives overlap heavily. In Fig. 8a, the decision boundary is indicated by the straight line. Note that, effectively, images in the "weak" zone were rejected without decision in [2]. Rather than reject such images outright, we attempt to utilize weak evidence in the overall probabilistic framework. We defer the explanation on how the decision boundary is determined until Section 4.1.

This technique is repeated on the edge direction histogram feature set in a similar manner. The result is shown in Fig. 8b.

4 High-Level Inference from 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, ceiling/wall,* and *grass* in order of decreasing usefulness, 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 to be useful, but its low occurrence and the variety of languages one needs to handle makes it unattractive.



(a)

(b)

Fig. 8. (a) Maximum SVM score versus difference between the top two scores for color moment features. Correctly-classified images are marked with triangles, while those classified incorrectly are marked with circles. See text for explanation. (b) Maximum SVM score versus difference between the top two scores for edge direction features. See text for explanation.

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. Detailed information regarding the detection of the selected semantic cues can be found in previously published papers on the related topics [9], [12].

4.1 Orientation by Face Detection

We detect human faces using an algorithm based on Schneiderman's original algorithm [7], with necessary

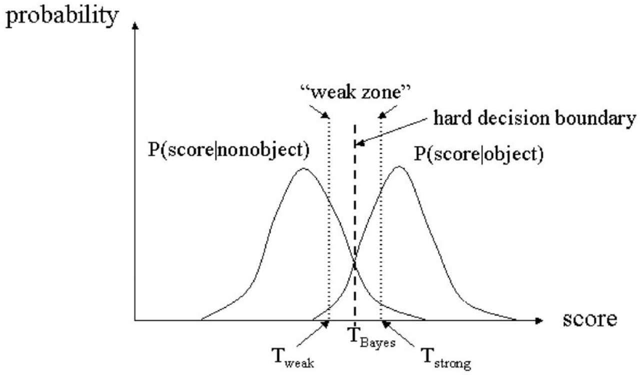


Fig. 9. A modified Bayesian classifier with a "weak zone" [T_{weak}, T_{strong}] defined around the Bayes decision boundary T_{Bayes}.

improvements to make the original algorithm more robust for unconstrained, consumer images. 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 the algorithm (e.g., feature extraction and model initialization) is only incurred once.

It is very important for each detected face to be associated with a confidence level in order to derive an optimal decision on the orientation of the image. The output of the face detector is a continuous-valued number akin to a probability with a higher score indicating stronger confidence. Based on the distributions of $P(\mathrm{score} \mid \mathrm{face})$ and $P(\mathrm{score} \mid \mathrm{nonface})$ obtained using an independent validation set of 600 images, we were able to determine thresholds, T_{strong} and T_{weak}, for declaring "STRONG," "WEAK," or "NO" face detection.

We chose to discretize the face detector output into three levels, taking the following considerations into account. First, we need a confidence value associated with the output of the face dectector. On one hand, it would be intractable if we used continuous output values or too many discrete levels. On the other hand, a binary decision with two levels would be unnecessarily harsh in an overall probabilistic inference scheme.

Second, using three levels is a natural extension of a classic Bayesian classifier. In Fig. 9, the probability distributions of both $P(\mathrm{score} \mid \mathrm{face})$ and $P(\mathrm{score} \mid \mathrm{nonface})$ are shown to be overlapping with each other. A Maximum Likelihood (ML) classifier would draw the decision boundary (i.e., $\mathrm{T_{Bayes}}$) where the two distributions intersect. However, there exists a "weak zone" around such a decision boundary, as shown between the two dotted lines, where the decision is murky at best. On the left side of this weak zone, it is mostly certain that the sample is not the object. On the right side of the zone, the confidence of seeing an object is high. The establishment of the weak zone naturally gives rise to "STRONG," WEAK," and "NO" detections. In essence, we attempt to introduce a soft decision by defining a "weak zone" around the Bayesian decision boundary where the two distributions overlap heavily. The width of the zone, i.e., $\mid T_{strong} - T_{weak} \mid$, was set to be proportional to the degree of overlap (e.g., 5 percent) between the two distributions. The ambiguity in the weak zone will be resolved by exploiting the majority of such weak votes, as opposed to being based strictly on the face detector output values, which are unreliable in such cases. Clearly, when multiple people are in the same image, the faces should, in general, be oriented in a consistent direction (otherwise, it would not matter). If such consistency does not exist, it is likely that false positive faces are detected and, therefore, the best decision is not to make orientation prediction based on the detected faces.

```
if score$_{max}$ >= $T_{strong}$
    label frame with the orientation that produces score$_{max}$ with STRONG confidence
else if 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
```

Fig. 10. Algorithm for inferring image orientation from orientation of detected faces.

Based on the above discussions, we label orientation using the pseudocode presented in Fig. 10.

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 percent exception). For portrait images, strong face-based orientation classifications were 99 percent correct. For consumer images, strong classifications were 90 percent correct while declining to label 55 percent of images; including weak classifications led to 81 percent accuracy with 42 percent of images unlabeled. However, faces alone would not solve the image orientation problem because approximately half of the images remain unlabeled. In addition, the face detector is far from being 100 percent correct.

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. The most prominent characteristic of sky is its color, usually light blue when it is clear. Cloudy/overcast sky tends to be white or neutral in color, while mixed sky has both clear blue sky and cloudy/overcast sky present in it. 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 "true" sky detection has not been used for image orientation to date (i.e., the "chicken and egg" problem). In fact, in the few attempts to do so, e.g., [19], the authors refrained from calling it sky detection (instead calling it "blue region detection") because water, for example, tends to be blue but is located at the bottom of the image (causing such location-based algorithms to fail).

In contrast, we have developed a physical model-based blue sky detector that 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 toward a distant horizon line in

```
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 x sqrt(area) with STRONG confidence

else if $score_{max} >= T_{weak}$
    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
```

Fig. 11. Algorithm for inferring image orientation from gradient of detected blue sky regions.

the image. The gradient in the sky, rather than the location of sky, naturally gives away the orientation of the image. Furthermore, the detector is extremely 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 detects large clear blue sky regions in two stages. In the first stage, a multilayer neural network, trained in a bootstrapping fashion, performs pixel classification based on color and texture features. 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, we infer the orientation of sky by examining vertical/horizontal gradients for each extracted region. Based on the estimated horizon-to-zenith direction, the algorithm then determines a probability to reflect how well the region fits the physics-based sky model. For more details, see [9].

The rest is straightforward, except that we need to account for the detection confidence levels, as well as multiple detected sky regions. The same procedure used for face was used to determine the optimal thresholds, T_{strong} and T_{weak}, for declaring "STRONG," "WEAK," or "NO" blue sky detection, as presented in Fig. 11.

Blue sky detection turns out to be even more reliable (96 percent 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 percent of the time).

One potential issue with using sky detection is that sky orientation may have been captured implicitly to some degree by statistical learning of the low-level feature-based method. We have investigated this issue using Bayesware Discoverer, a Bayesian network analysis package based on the methodology by Cooper and Herskovits [15]. 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).

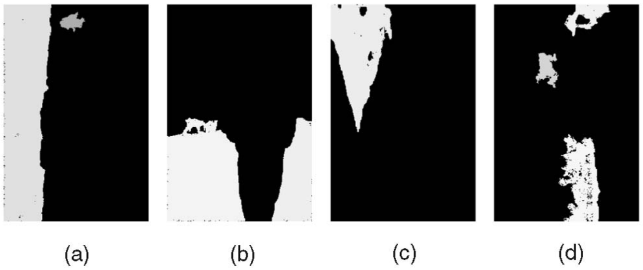


Fig. 12. Orientation determined by the spatial configuration of detected **grass** regions. (a) E(1.0). (b) N(1.0). (c) E(0.6) or S(0.4). (d) E(0.5) or W(0.5). (a) and (b) are straightforward. (c) points most likely to the *east* (right side) and also likely to the *south* (bottom side). (d) points to either *east* or *west* (left or right sides).

4.3 Orientation by Grass Detection

While image orientation can be directly inferred from face and blue sky, only the spatial configuration of grass region(s) provides indication of the most plausible image orientation. Classifiers need to be designed carefully to infer image orientation from typical spatial configurations, such as those shown in Fig. 12. The corresponding original images are deliberately not shown so readers can attempt to infer the most likely image orientation based *solely* on results of grass detection. This will also be the case with cloudy sky and ceiling/wall. 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 grass region in Fig. 12d, it is equally likely that the orientation is east or west.

A neural network, similar to that used for detecting clear blue skies, is used for performing the pixel-level classification. The input to the neural network classifier per pixel consists of the three color features and six texture features (coefficients from a 2-level wavelet transform). In particular, green foliage regions (trees) are used as negative examples to bootstrap the neural network 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. After the pixel classification, connected components are extracted as grass regions. A few typical spatial configurations of detected (potential) grass regions are shown in Fig. 12.

We extracted the following salient features from the grass region images:

- Percentage of image area occupied by the grass regions (1).
- Percentage of each border occupied by the grass regions (4).
- Aspect ratio of the bounding box containing all the major grass regions (1).
- Ratio of distances to the two sides parallel to the bounding box's major axis (1).

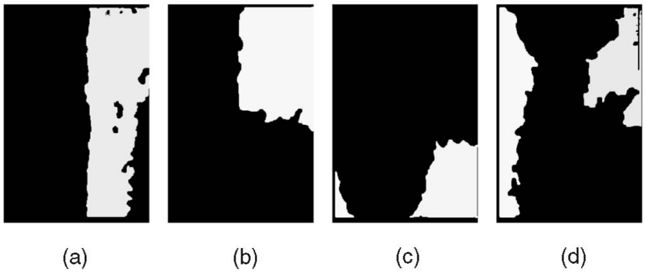


Fig. 13. Orientation determined by the spatial configuration of detected **cloudy sky** regions. (a) E(1.0). (b) N(0.5) or E(0.5). (c) S(1.0). (d) none. (a) and (b) are clear. The only possible explanation for (c) is *south* (bottom-up), while, in (d), the opposite locations of the two regions and the fact that one region extends all the way along one border leads to "no decision."

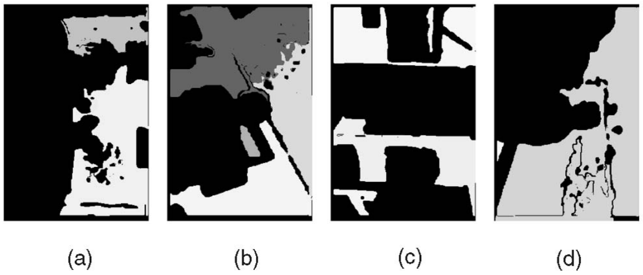


Fig. 14. Orientation determined by the spatial configuration of detected white **ceiling/wall** regions. (a) E(1.0). (b) E(1.0), (c) none, (d) none. (a) is simple (east is the top), but (b) is tricky; three sides filling up makes *east* (right side) the only plausible top. It is clear that no decision can be made for (c), while careful consideration also leads to a "no decision" in (d).

These features (seven in total) served as input to another multilayer neural network (different from the earlier one used for pixel classification). The network has five output nodes, one for each orientation and one for "unknown," and two hidden layers of five nodes. We trained the network on an independent set of 936 consumer images, each of which were assigned "ground truth" belief values for the four possible orientations by two observers inspecting the detected grass regions; ambiguous spatial configurations and images with little (e.g., under 1 percent of the image) or no grass were assigned an "unknown" label.

4.4 Orientation by Cloudy Sky Detection

Cloudy sky is also a useful cue for image orientation. 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.

We have observed that cloudy or overcast sky regions tend to be the brightest regions in an image because sky is almost always the main source of illumination in outdoor scenes. We convert the image to LUV space 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., $\mathrm{L}' = \mathrm{L}/\mathrm{L}_{\max}$, where $\mathrm{L}_{\max}$ is the maximum raw luminance over the entire image. This physics-motivated feature, though less rigorous, leads to significant reduction in misclassifying other grayish colored subject matter as cloudy/overcast sky.

The cloudy sky pixel classification is performed using the same methodology as that for grass, except that the color features are $(\mathrm{L}', \mathrm{U}, \mathrm{V})$. The process of inferring image orientation from detected cloudy sky regions is also identical to that used for grass regions (see Fig. 13 for examples).

4.5 Orientation by Ceiling/Wall Detection

The detector for white or off-white ceiling/wall is essentially the same as the cloudy sky detector because of the similar color and texture characteristics. The difference is that a higher degree of occlusion often occurs with the wall and ceiling (often connected) and more straight lines may be present. A few examples are shown in Fig. 14. We used the same methodology as for cloudy sky to detect ceiling/wall regions and infer image orientation from them.

4.6 Alternative Approach

We experimented with an alternative means of inferring image orientation from spatial configurations. The motivation was to process the data as little as possible, potentially to learn patterns not obvious to algorithm designers. To that end, low-resolution (16×24) thumbnails of the material detection maps were used. We applied PCA to these low-resolution images to further reduce dimensionality. The first 12 principal components of grass regions computed from the training set described in Section 4.4 are shown in Fig. 15. Clearly, the first components ("eigengrass") correspond to canonical grass locations. We then classified the 12D features using a multilayer neural network with outputs like those described above (i.e., four orientations and "unknown"). Not surprisingly, even with various hidden layers, this approach performed worse than the feature-based approach, except for the obvious cases where a grass or cloud region spans one whole side of an image. Such a holistic classifier was unable to capture more subtle properties, such as whether the material region touched the

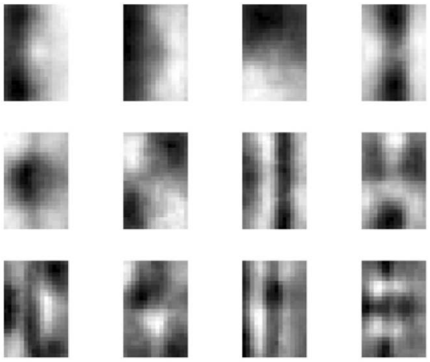


Fig. 15: The first 12 principle components of grass regions ("eigengrass"). Clearly, the first components correspond to canonical grass locations.

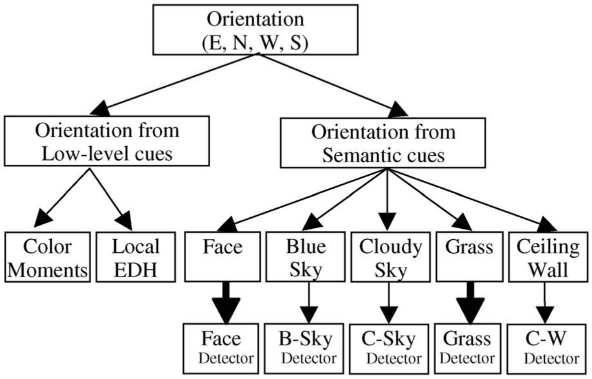


Fig. 16. Structure of the Bayes network.

border of the image and its degree of elongation. This is evident from the principal components.

5 Confidence-Based Cue Integration

Whether using low-level or semantic cues for prediction, the predictors may or 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 Bayesian network (BN) [11]. Because the predictors differ in their nature, occurrence, and confidence, a statistically optimal way is to combine them in the probability domain (versus a monolithic feature vector). A Bayesian classifier according to the *maximum a posteriori* (MAP) criterion gives orientation $\omega \in \{\mathrm{N}, \mathrm{E}, \mathrm{S}, \mathrm{W}\}$ by:

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

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

Determining the structure of the BN is straightforward once various conditional independencies between various cues are factored based on domain knowledge (e.g., the orientation estimates derived from the semantic cues are independent of each other given the true orientation of the image). The BN structure is shown in Fig. 16. Note that we deliberately separate the actual detectors so that any improvement in a detector can be readily incorporated without retraining the network; only the associated detector's confusion matrix needs to be replaced at the bottom level of the network.

TABLE 1
The CPM of *Face* Evidence

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

See text for abbreviations.

TABLE 2
The CPM of *Grass* Evidence

GE	GN	GW	GS	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

TABLE 3
The CPM Connecting *Face* to *Orientation from Semantic Cues*

FE	FN	FW	FS	ND	
0.75	0.02	0.01	0.02	0.20	SemanticsEast
0.02	0.75	0.02	0.01	0.20	SemanticsNorth
0.01	0.02	0.75	0.02	0.20	SemanticsWest
0.02	0.01	0.02	0.75	0.20	SemanticsSouth
0.001	0.001	0.001	0.001	0.996	NoSemantics

The parameters of the Bayes network, i.e., the individual conditional probability matrices (CPMs), were obtained from training. Three examples of the CPMs are included below in Tables 1, 2, and 3. 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 faces, a strong cue (the first highlighted edge in Fig. 16). The second example is related to grass, a weak cue (the other highlighted edge). The third example is related to "orientation from semantic cues," the intermediate node handling the various semantic cues. Note that these semantic cues are only conditionally independent. Therefore, we need an intermediate, nonevidence node because given the correct image orientation, the orientations inferred from semantic cues become independent. For example, the first row of $\mathrm{P(Face \mid Orientation\ from\ Semantic\ Cues)}$ means, given that image orientation is east and there are semantic cues (any of the ones on our list) present, the probability of having "Face East" is 0.75, "Face North" is 0.02, "Face West" is 0.02, "Face South" is 0.02, and the probability of seeing no faces is 0.20. 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.

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 scanned from film,

TABLE 4
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/wall, and cloudy sky	**89.9%**	**82.5%**
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%

**Results shown for individual color and edge direction features were determined directly not using the Bayes net, hence there are no results given incorporating class priors.*

extensive studies showed that the priors of the four orientations are roughly [3]:

$$\text{East } 0.72 \quad \text{North } 0.14 \quad \text{West } 0.12 \quad \text{South } 0.02.$$

6 Experimental Results

We conducted extensive experiments to illustrate the performance of the proposed algorithm under various configurations. The low-level classifiers were trained on a mixture of 4,572 Corel images (1,136 per class) and 3,744 consumer images (936 per class). The semantic detectors were trained on different sets of images completely unrelated to this study.

Our independent testing set consists exclusively of a separate set of 3,652 consumer images (913 per class). 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 photographs (testing set only) are given in Table 4 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.

In order for the system to be viable, the minimum goal for an automatic algorithm is to beat the prior, which is, as previously stated, 72 percent 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 percent when only low-level cues ("CM+EDH") are used.

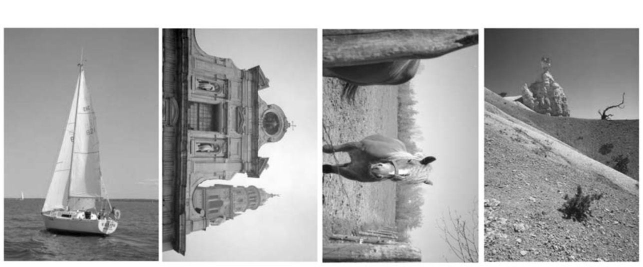


Fig. 17. Examples for which the low-level feature-based approach is effective.

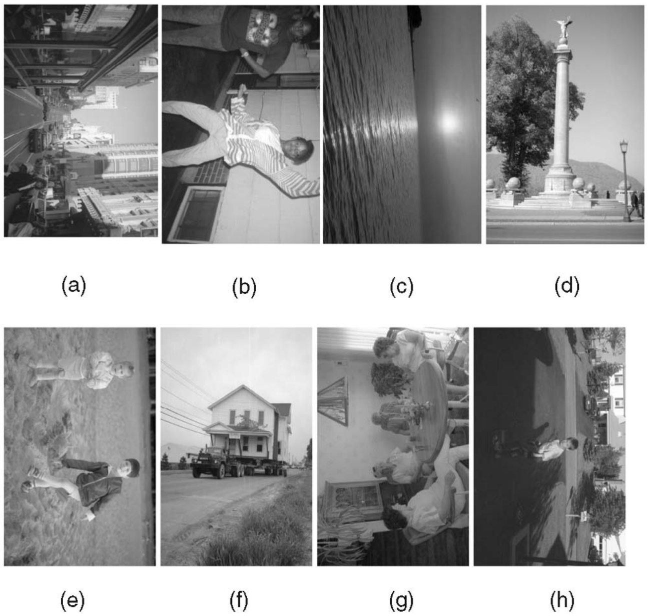


Fig. 18. Examples for which the integrated approach is successful. The images are shown in their original orientation. (a) L:Sw-, SB:Es+, (b) L:Ss-, F:Es+, (c) L:Es+, NS, (d) L:Es-, SB:Ns+, (e) L:Nw-, F:Es+, (f) L:Nw+, SC:Ns+, G:NW, (g) L:W+, NF, and (h) L:Ew+, F:Es+, G:EW. The notations are: "L" for 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; "E/N/W/S" for the four orientations of *East, North, West,* and *South*; "w" for weak prediction and "s" for strong prediction; "+" for correct prediction and "-" for incorrect prediction.

Face detection added significant gains in accuracy: 7.3 percent with equal prior and 5.3 percent with the true prior. Note that these are additional gains. The 3.4 percent gain from adding blue sky is significant, given that the "easy" cases (e.g., those in Fig. 17) 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., Fig. 18a) or at an unusual location (e.g., Fig. 18d). Note that the effect of blue sky is less pronounced (1 percent) with the true prior because blue sky tends to appear more often in landscape images.

The remaining, weaker semantic cues added approximately another 1 percent in accuracy, more so (1.5 percent) 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. The overall accuracy for the fully integrated algorithm is 90 percent.

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

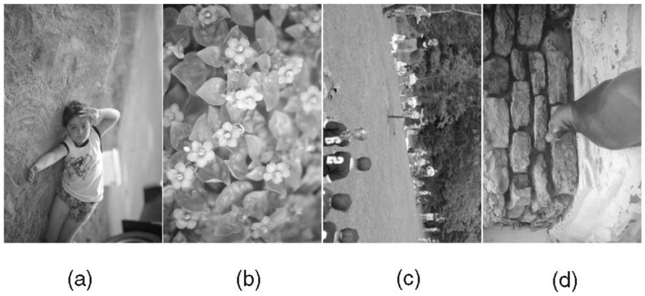


Fig. 19. Examples of the failures of the integrated approach. These failures were due to concept failure ((a) face of a person lying down), false positive detection ((b) and (c) faces), and no semantic cues when low-level cues are incorrect ((d)).

values of the Bayes network. Note that this is even more useful in the equal prior case (Table 4, last row).

A number of examples are included to demonstrate the effects of the proposed system. Fig. 17 contains examples when 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 Fig. 18 where semantic cues either override the incorrect classification by the low-level predictors (Figs. 18a, 18b, 18d, and 18e) or strengthen correct but perhaps weak predictions (Figs. 18f and 18h). In particular, the low-level cues predicted "south" for Figs. 18a and 18b because the tops of these images are darker and more textured; "east" for Fig. 18d because the sky region is predominately on that side; and "north" for Fig. 18e because the person wearing darker clothes is at the bottom. Strong faces (Figs. 18b, 18e, and 18h), strong blue sky (Figs. 18a and 18d), and strong cloudy sky (Fig. 18f) were the winning factors, while grass helped limit the possible orientations to two out of four orientations (Figs. 18f and 18h).

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 (Fig. 18c) and no face was detected for the small and turned heads in (Fig. 18g).

We also analyzed the failure cases of the integrated system. Some typical examples are shown in Fig. 19. These failures were due to concept failure (Fig. 19a: face of a person lying down), false positive detection (Figs. 19b and 19c: faces), and no semantic cues when low-level cues are incorrect (Fig. 19d).

The computation time on a SunBlade-1000 workstation is approximately 6 seconds per image, with about 5 seconds devoted to low-level feature extraction and classification using unoptimized code and the remaining time spent primarily on semantic feature extraction (optimized code) and orientation prediction; the Bayesian network uses negligible time.

7 Discussions and Conclusions

This work represents an attempt to solve an intrinsically high-level vision problem such as image orientation detection using both low-level vision features and selected semantic cues that are inspired by a psychovisual study [4] 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 decision tree starting with the strongest predictors [6]. This only works if all predictors are fairly strong and the strongest one has 90+ percent accuracy, otherwise the overall accuracy will be below 90 percent (the level achieved by our integrated approach). While our face and blue sky detectors are robust *and* image orientation can be robustly inferred from them, all 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 (computable) cues and the priors, our current accuracy is 90 percent for unconstrained consumer photos; without taking advantage of the priors, the accuracy is 83 percent, approaching the average of humans when viewing thumbnails. The proposed framework is a successful attempt to bridge the gap between computer and human vision systems and is applicable to other scene understanding problems [20].

8 Future Work

With the recent advance in spatial context-aware semantic material detectors [12] and expected added capability in detecting profile faces, it is possible that inconsistencies among various otherwise independent semantic detectors can be alleviated and missed opportunities can be retrieved to lead to higher overall accuracy of the integrated orientation detection algorithm. As for other semantic cues, such as *snow, water, trees, animals* (all species), *buildings* (all types and styles), *ground* (pavement, sand, dirt, carpet, floor, etc.), *furniture* (all types), and *vehicles* (all types and models), the issue we are facing is diminishing returns; it not only becomes far more difficult to develop robust algorithms for detecting these less well-defined types of objects, but they also appear less frequently in images and have weaker correlation with the correct image orientation [4]. In addition, these cues may be redundant for the purpose of determining the image orientation.

Appendix

For examples of the consumer photos used in this study, please see Fig. 20.

Acknowledgments

The authors would like to thank Henry Nicponski for his help with using face detection for orientation determination, Amit Singhal for lending his expertise on Bayesian networks, and Robert T. Gray for valuable discussions.

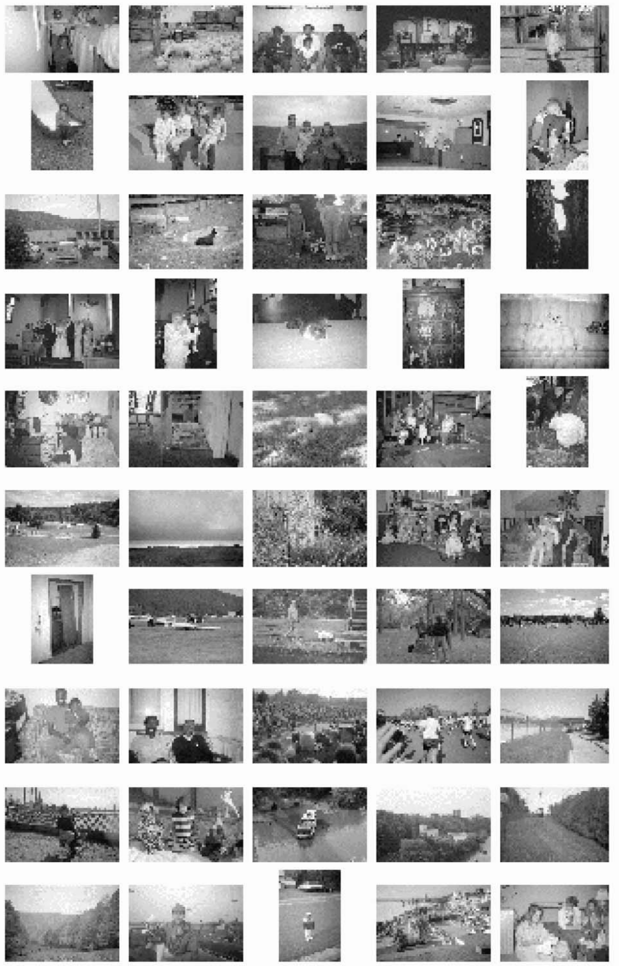


Fig. 20. Subset of the consumer photos used in this study.

REFERENCES

[1] A. Vailaya, H.J. Zhang, and A. Jain, "Automatic Image Orientation Detection," *IEEE Trans. Image Processing,* vol. 11, no. 7, pp. 746-755, 2002.

[2] Y. Wang and H. Zhang, "Content-Based Image Orientation Detection with Support Vector Machines," *Proc. IEEE Workshop Content-Based Access of Image and Video Libraries,* 2001.

[3] R. Segur, "Using Photographic Space to Improve the Evaluation of Consumer Cameras," *Proc. IS&T Image Processing, Image Quality, Image Capture and Systems (PICS) Conf.,* 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, pp. 429-457, 2003.

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

[6] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification.* New York: John Wiley & Sons, 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 Univ., 2000.

[8] A. Vailaya and A. Jain, "Detecting Sky and Vegetation in Outdoor Images," *Proc. 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 Trans. Image Processing,* vol. 11, no. 3, pp. 201-212, 2002.

[10] R.P.W. Duin, "The Combining Classifier: To Train or Not to Train?" *Proc. Int'l Conf. Pattern Recognition,* 2002.

[11] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* San Francisco: Morgan Kaufmann Publishers, Inc., 1988.

[12] A. Singhal, J. Luo, and W. Zhu, "Probabilistic Spatial Context Models for Scene Content Understanding," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* 2003.

[13] C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery,* vol. 2, no. 2, pp. 121-167, 1998.

[14] D.M.J. Tax and R.P.W. Duin, "Using Two-Class Classifiers for Multiclass Classification," *Proc. Int'l Conf. Pattern Recognition,* 2002.

[15] G. Cooper and E. Herskovits, "A Bayesian Method for the Induction of Probabilistic Networks from Data," *Machine Learning,* vol. 9, pp. 309-347, 1992.

[16] J.R. Smith and C.-S. Li, "Image Classification and Querying Using Composite Region Templates," *Computer Vision and Image Understanding,* vol. 75, nos. 1-2, pp. 165-174, 1999.

[17] Z. Tu and S.-C. Zhu, "Image Segmentation by Data-Driven Markov Chain Monte Carlo," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 5, pp. 657-673, May 2002.

[18] D. Comaniciu and P. Meer, "Mean Shift: A Robust Approach toward Feature Space Analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 5, pp. 603-619, May 2002.

[19] R. Goodwin, *Whole Order Orientation Method and Apparatus,* US Patent #5642443, 1997.

[20] J. Luo, A. Singhal, S.P. Etz, R.T. Gray, "A Computational Approach to Determination of Main Subject Regions in Photographic Images," *Image Vision Computing,* vol. 22, no. 3, pp. 227-241, 2004.

Jiebo Luo received the BS degree in electrical engineering in 1989 from the University of Science and Technology of China and the PhD degree in electrical engineering from the University of Rochester, Rochester, New York, in 1995. In the summer of 1995, he was employed at the Joseph C. Wilson Center for Technology of Xerox Corporation, Webster, New York. In 1995, he became a senior research scientist and is currently a senior principal scientist in the Research and Development Laboratories, Eastman Kodak Company, Rochester, New York. His responsibilities include leading research and development of intelligent algorithms for current and future digital imaging products. He has authored more than 100 technical papers and holds more than 30 granted US patents. Dr. Luo was the recipient of the Best Student Paper Award for Visual Communication and Image Processing from SPIE in 1994 and a Certificate of Merit for Scientific Exhibit from RSNA in 1998. He was the chair of the Rochester section of the IEEE Signal Processing Society in 2001 and the general cochair of the IEEE Western New York Workshop on Image processing in 2000 and 2001. He was also a member of the organizing committee of the 2002 IEEE International Conference on Image Processing, a guest coeditor for the *Journal of Wireless Communications and Mobile Computing* special issue on multimedia over mobile IP, the lead guest editor for the *Pattern Recognition* special issue on image understanding for digital photos, and a member for the Kodak Research Scientific Council. Currently, he is an associate editor of the journals *IEEE Transactions on Multimedia*, *Pattern Recognition* and the *Journal of Electronic Imaging*. Dr. Luo is an adjunct professor at the Rochester Institute of Technology and has served as a coadvisor or thesis committee member of many PhD and MS graduate students in various universities. His research interests include image processing, pattern recognition, computer vision, medical imaging, and multimedia communication. He is a Kodak Distinguished Inventor and a senior member of the IEEE.

Matthew Boutell received the BS degree in mathematical science from Worcester Polytechnic Institute, Massachusetts, in 1993 and the MEd degree from the University of Massachusetts at Amherst in 1994. He served for several years as a mathematics and computer science instructor at Norton High School and at Stonehill College. Currently, he is a PhD student in computer science at the University of Rochester. His research interests include pattern recognition, probabilistic modeling, and image understanding. He is a student member of the IEEE Computer Society.