Content-based image orientation recognition

Ekaterina Tolstaya
Image Enhancement Technology Group, Printing Technology Lab
Samsung Research Center, Moscow, Russia
e.tolstaya@samsung.com

Abstract

In this paper a method for digital image orientation recognition is proposed. Feature vectors are chosen to be flip-invariant to effectively classify images onto portrait-oriented and landscape-oriented. A new texture feature is proposed based on the observation that more textured areas are located usually in the lower part of the image. The method was implemented in software and tested using an image set containing various photo images.

Keywords: *image orientation, classification, boosting, texture features.*

1. INTRODUCTION

Digital cameras have gained a great popularity, and as a result a huge volume of digital images is produced. However, when the images are viewed they may not always be displayed in their preferred viewing orientation. The preferred viewing orientation is the orientation in which the image was captured.

Displaying images in their correct orientations is necessary in various image processing applications. For example, it is required in image albuming [15] and display, as well as in auto-collage applications [14], also it is an important step in face detection methods, since a lot of them require predefined face orientation. While manually adjusting orientations for several images is trivial, it is more efficient to be able to automate on several hundred digital photographs taken from a field trip or a vacation. One solution is to have the digital cameras record, at the time of capture, the orientation information in the image file EXIF tag. However, very often this information is absent, or is incorrect. A more practical alternative then is to design systems that are able to determine image orientations using image analysis.

2. CONTENT BASED ORIENTATION RECOGNITION TECHNIQUES

Technically, the goal of automatic image orientation recognition is to classify an image to one of the four possible orientations, corresponding to rotation angles of 0º, 90º, 180º and 270º. Nevertheless, in practice, it is usually sufficient to determine if an image is not rotated (0º orientation) or rotated counter-clock-wise (CCW) or clock-wise (CW) (90º or 270º orientation correspondingly), as it is rare that a picture is taken upside down.

Automatic image orientation recognition is a relatively new research area in computer vision. Most of the early work focused on documents, and success was largely due to the constrained nature of the problem (text cues). For natural images, the problem is considerably more challenging. Until recently ([1, 3]), there had been little work on automatic image orientation recognition for natural images. Humans appear to 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 this way because current object recognition algorithms are extremely limited in their scope and robustness. Out of millions of possible objects that can appear in a natural scene, robust algorithms exist for only a few object categories (such as face, sky). To date, scene classification is often approached by computing low-level features (such as color, texture, and edges) that are processed with a learning engine to directly infer high-level information about the image ([1, 3, 12]). Recently, a new approach was proposed that combines low-level features with detectable semantic scene content in order to improve the accuracy of indoor-outdoor image classification ([13]).

Existing automatic image orientation recognition methods fall into two main categories. Top-down methods are based on high level perception cues (i. e. the detection of faces, sky and walls [8]), or semantic relations in image contents (i. e. textured area in lower part [4]). However, top-down methods suffer from the instabilities of current object detection and recognition algorithms, and are more likely to bias to a particular set of training images. On the other hand, bottom-up methods determine image orientations with low-level features; examples include color moments [3] and edge direction histograms [1, 2]. Compared to high-level cues, low-level features are more robust and reliable.

A comprehensive study on psychological aspects or image orientation recognition was presented in [9]. The experiment reported in [9] investigates the perception of orientation of color photographic images. A collection of 1000 images (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. The results show that for typical images, accuracy is close to 98% when using all available semantic cues from high-resolution images, and 84% when using only low-level vision features and coarse semantics from thumbnails. Study also revealed that most useful and reliable cues used by humans at various image resolutions are sky, people, color, texture and trees and water.

However, while humans recognize thousands of objects and use them to make complex inferences about orientation, robust detection algorithms exist only for a handful of objects. Close-up images, low-contrast images, or images of uniform or homogeneous texture (e.g., sunset/sunrise and indoor images) pose serious problems for robust orientation estimation.

Psychophysical studies in [9] also confirmed that low-level features are critical for human performance on determining image orientations.

3. RELATED WORK

The majority of works devoted to content-based automatic image orientation recognition use learning techniques for image orientation recognition. In [3] a comparison between different classifiers is presented, namely *k*-nearest neighbors, support vector machine (SVM), a mixture of Gaussians, and hierarchical discriminating regression (HDR) tree. Authors of [3] demonstrate that best accuracy is achieved by SVM, however, when empowered with LDA (linear discriminant analysis) method all methods significantly improve their performance, and the highest accuracy is achieved by mixture of Gaussians (with LDA).

Usually an image is divided into NxN blocks and features are extracted from those fragments. In [3] N=10 is suggested, in [8] authors claim that N=7 is enough, in [1] N=8. Empirical knowledge suggests that the essential color information for orientation recognition is usually embedded in the periphery rather than the central part of the image. Based on this idea, authors of [1] take into account only peripheral image blocks. The majority of authors prefer using the LUV color space instead of RGB, however, in [5] several different color spaces are used (RGB, YIQ).

Features used for orientation recognition typically include color moments (CM), such as component-wise mean and variance within each block, along with edge features, namely edge directions, edge direction histograms, quantity of edge pixels, etc., computed from the lightness component.

In [8] and [4] both low-level image features, as well as semantic cues are jointly used in image orientation recognition. Results reported by authors are 84-94% in [4] (depending on cue, and overall 94%) and are 70-82% (without rejection) in [8].

Rejection is widely employed by many authors. It means that when orientation recognition confidence is low, the image is marked as “Not Detected”. Low confidence is typical for close-up views, uniformly textured images and nearly diagonal rotations. Usually it is defined using small absolute values of SVM output probabilities, and confirmed by setting some threshold. If the probability is smaller then threshold, orientation is not detected. In [1] and [6] for testing confidence image is rotated by 180, and if detected orientation of rotated image disagrees with orientation detected earlier, image is rejected (see Figure 1).

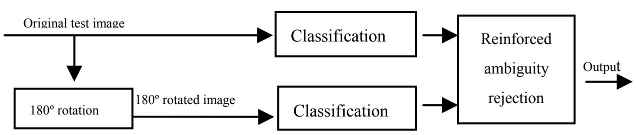

Figure 1. Re-enforced ambiguity rejection from [1] and [6].

In [1, 6] delivered accuracy lies between 78% and 96%, depending on classifier architecture (single or double layer), and rejection scheme. Authors claim that rejection scheme in their case does not significantly improve the results (within 1%).

In [7] images are first classified onto portrait/landscape orientation (90º/270º or 0º/180º), and then classified on four possible orientations.

In [6] accuracy reported by authors varies from 78% (with 0% rejection) to 96.5% (with 50% rejection). In [5] accuracy varies from 27% to 96%, depending on image type, the worst case being backgrounds and close-ups.

In [2] classification method is boosted by indoor/outdoor classification.

4. IMAGE ORIENTATION RECOGNITION

It seems natural to first classify images on portrait/landscape orientation, since greater part of photos are taken in landscape orientation, among them upside-down rotated photos (180º) are found very rarely. Further discriminate between 90º/270º rotated images is easier than between four possible rotations.

Proposed orientation technique is based on application of AdaBoost based on weighted voting of elementary classifier committee [10], which classify image orientations according to features, extracted from images. There are many potential features which can be used to represent an image. Different features have different abilities to detect if an image is portrait or landscape oriented. Since global image features are not invariant to image rotation, we prefer to rely on local regional features for classification.

4.1 Luminance and chrominance features

First, luminance and chrominance features are extracted. Image of the size M by N pixels is converted to YCrCb color space and then divided by S *horizontal* blocks, such that width of the block is equal to the width of the image M, and height constitutes a N/S fraction of the image height. As it was noted, with growing number of image fragments classification accuracy increases, but complexity learning and classification processes increases significantly.

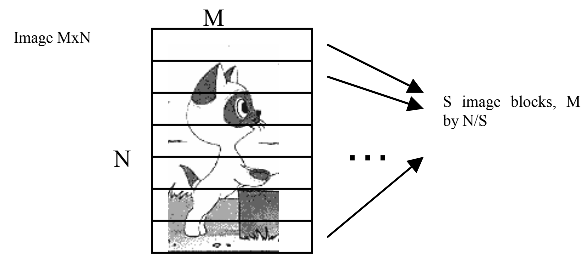

Figure 2. Scheme of image blocks, width of the block is equal to image width; height constitutes a N/S fraction of the image height.

In such a way for every color component, a feature is represented as a vector, its components are chrominance or luminance characteristics, $\{x_1 ... ,x_{S*K}\}$, where K is the number of characteristics. The following characteristics of each fragment were computed:

$$\left.\begin{aligned} x_{(p-1)K+1} &= \frac{S}{MN}\sum I(i,j), \\ x_{(p-1)K+2} &= \sqrt{\frac{S}{MN}\sum [I(i,j)-x_{(p-1)K+1}]^2}, \end{aligned}\right\},$$

where $i \in [1,..,M]$, $j \in \left[(p-1)\frac{N}{S},..,p\frac{N}{S}\right]$, p=[*1..S*]. Feature vector is formed by 3 vectors of each color component. This vector is invariant to image flipping relative to vertical axes. Then image is divided into S *vertical* blocks, such that height of the block is equal to the of the image height N, and width constitutes a M/S fraction of the image width, and all the procedure of computing features is repeated.

4.2 Texture feature

Texture feature is computed using RGB color space. First, as previously, image is divided by S horizontal blocks, such that width of the block is equal to the width of the image M, and height constitutes a N/S fraction of the image height. For texture feature S is significantly smaller than for color feature. Mean values of *R*, *G*, and *B* channels are computed, R^m, G^m, and B^m correspondingly. Then for each pixel inside image block an angle is computed according to the following formula:

$$A(i,j) = \left| \cos^{-1} \frac{R_{ij}R^m + G_{ij}G^m + B_{ij}B^m}{\sqrt{R_{ij}^2 + G_{ij}^2 + B_{ij}^2}\sqrt{R^{m2} + G^{m2} + B^{m2}}} \right|$$

A histogram of angle distribution is constructed. All angles are clipped to [0, π/4], because R, G, B ≥ 0, and the majority of angles absolute values are smaller than π/4. Figure 3 provides illustration of the described texture feature.

Figure 3: Photo (upper panel) and image of angles (photo is divided into 50 horizontal blocks, each pixel of the left image is computed according to formula above), in logarithmic scale

Angle histogram (AH) is generated for each of the S regions to characterize image structural and texture information. This choice is based on the observation that generally more texture is present in the lower rather than the upper part of the image.

This feature vector is invariant to image flipping relative to vertical axes. The array of edge directions is divided into S vertical blocks, such that height of the block is equal to the of the image height N, and width constitutes a M/S fraction of the image width, and all the procedure is repeated.

4.3 Classification system

The classification system is a two-class classifier. For the case of separating the set of training vectors belonging to two classes, let $\{x_1; y_1\}; \ldots; \{x_m; y_m\}$ denote a set of training data, where $x_i \in R^N$ is a feature vector and $y_i \in \{-1,+1\}$ is its class label. Classification is application of Real AdaBoost based on weighted voting of elementary classifier committee [10]. Decision function is constructed as a weighted sum of elementary classifiers:

$$F(x) = sign\left[\sum_{m=1}^{M} w_m h_m(x)\right].$$

Elementary classifier here consists of comparison of corresponding feature vector component to some threshold value, computed in advance at training stage. Function *F(x)* assigns input test feature vector *x* to one or another class.

There are several AdaBoost algorithms which differ by approaches for optimization of weights w_m. In some realizations of these algorithms it is possible to adjust parameters of simple weak learners, in particular to optimize thresholds. We used the GML AdaBoost Matlab Toolbox, building of classifiers committee and adjusting parameters of weak learners. GML AdaBoost Matlab Toolbox (http://research.graphicon.ru/) is a set of Matlab functions and classes implementing a family of classification algorithms known as Boosting. Real AdaBoost is the generalization of a basic AdaBoost algorithm first introduced by Fruend and Schapire in [11].

A set of 800 images with known portrait/landscape orientation was selected for the training stage. Images were selected to come from various sources, including photos, captured by professional and amateur photographers, from different parts of the world, to prevent classification model to concentrate on some particular color combinations. Images were resampled to 200x200 pixels by nearest neighbor interpolation. Features were extracted from all images in the set. Number of images within each class was equal to 400.

For testing a set of 861 images was chosen, again from various sources. Then each image was assigned random orientation, 0º, 90º, or 270º; then it was rotated accordingly, and after that image features were extracted. Partition of portrait/landscape oriented images was kept about 50%. Scene types were distributed as follows:

Clear blue sky	87
Cloudy blue sky	122
Overcast sky	200
Sunset, sunrise	36
Night scenes	23
No or small portion of sky	393

Every image from the testing set was processed according to the following procedure: first, image was classified as landscape or portrait-oriented. In case it was portrait-oriented, first it was classified rotated by 90º or 270º, and probability p_1 was computed. 90º/270º rotation scheme was improved by additionally rotating the images by 180º, and subsequent 90º/270º classification and p_2 evaluation. Final probability is found as

$$p = \begin{cases} p_1, & if \quad |p_1| \geq |p_2| \\ -p_2, & if \quad |p_1| < |p_2| \end{cases}$$

This improved classification scheme provided a significant gain of accuracy. Note that 180º rotations can be simply implemented as feature vector flipping.

4.4 Rejection scheme

Probability distribution, which is the output of classifiers, measures the level of confidence in detected orientation. A classifier without rejection will assign some orientation even if it is not confident in its choice, i.e. when the probability is close to zero. But the majority of misclassification occurs exactly in these cases. So, it seems to be reasonable not to assign orientation, when confidence is low and, therefore, to reject an image from classification. As it was shown in [1] and [6] complex rejection schemes do not provide significant gain is classification accuracy, the gain being within 1%. In our work, the rejection scheme is based on a preselected threshold T: the image is portrait oriented, if $P<=T$, and landscape oriented, if $P>1-T$. Rejection rate shows, how many images were rejected, relatively to the image test set.

4.5 Results

Classification results on the training set are the following: 94% for the first classifier (portrait/landscape); and 87% for the second classifier (90º/270º rotation). Overall figures for classification scheme are presented in table below (number of images in test set is equal to 861). Here classification accuracy is computed as the number of correctly oriented images which were not rejected, divided by the number of not rejected images.

All (861 images)			
Rejection rate	0.4% (4)	13.7% (118)	40% (338)
Accuracy	87% (749)	88.5% (657)	90% (473)

Table 1. Classification results for portrait/landscape oriented images (actual number of images is given in brackets).

Rotated (586 images)			
Rejection rate	0.3% (2)	15% (92)	47% (275)
Accuracy	77% (451)	79.5% (392)	83% (259)

Table 2. Classification results for 90º/270º (actual number of images is given in brackets).

5. CONCLUSION

A method for image orientation recognition was developed, trained and tested. New texture feature vector was proposed and fast and efficient classification method was applied.

In future we plan to compare classification performance of new image features with old ones, as well as to compare traditional image division by blocks with proposed, where block width is equal to the image width. Also it is planned to apply LDA to feature sets to improve classification accuracy and to compare AdaBoost with SVM classification. Preliminary results showed that AdaBoost classification performance is similar to SVM, requiring, however, less storage space for weak learners, then for support vectors.

Proposed method is disclosed in patent application [16].

6. REFERENCES

[1] Wang, Y., Zhang, H., 2001, "Content-Based Image Orientation Detection with Support Vector Machines" in IEEE Workshop on Content-based Access of Image and Video Libraries, pp 17-23.

[2] Zhang, L, Li, M., Zhang, H, 2002, "Boosting Image Orientation Detection with Indoor vs. Outdoor Classification", Workshop on Applications of Computer Vision 2002.

[3] Vailaya A., Zhang, H., Yang, C., Liu, F., and A. K. Jain, 2002 "Automatic image orientation detection", IEEE Tr. On Im. Proc., vol. 11, No. 7

[4] Wang L., Liu X., Xia L., Xu G., Bruckstein A., 2003, "Image orientation detection with integrated human perception cues (or which way is up)" Proceedings of Int. Conf. on Image Processing, vol. 3, pp 539-542

[5] Baluja, S.; Rowley, H. A., 2005, "Large scale performance measurement of content-based automated image-orientation detection", 2005. IEEE Int. Conf. on Image Processing, vol. 2, pp. 514-517

[6] Wang Y., and Zhang H., 2004, "Detecting image orientation based on low-level visual content", Computer Vision and Image Understanding, pp. 328–346

[7] Siwei Lyu, 2005, "Automatic Image Orientation Determination with Natural Image Statistics", Proceedings of the 13th annual ACM Int. Conf. on Multimedia , pp. 491 – 494

[8] Luo, J., and M. Boutell, 2005, "Automatic Image Orientation Detection via Confidence-Based Integration of Low-Level and Semantic Cues", IEEE Trans. On Pat. Analysis and Machine Int., vol. 27, No. 5

[9] Luo J., Crandall D., Singhal A., Boutell M. and Gray R. T., 2003 "Psychophysical study of image orientation perception", Spatial Vision, Vol. 16, No. 5, pp. 429– 457

[10] Friedman J., T. Hastie, R. Tibshirani, 2000, "Additive logistic regression: A statistical view of boosting". The Annals of Statistics, 38(2):337–374.

[11] Freund, Y., R. Schapire, 1996 "Experiments with a new boosting algorithm", Int. Conf. on Machine Learning

[12] Szummer, M., and R. Picard, "Indoor-Outdoor Image Classification", IEEE International Workshop in Content-Based Access to Image and Video Databases, Bombay, India, Jan. 1998.

[13] Luo, J. and A. Savakis, "Indoor vs. Outdoor Classification of Consumer Photographs," Int. Conf. Image Proc. ICIP'01, Thessaloniki, Greece, Oct. 2001.

[14] Rother, C., Bordeaux, L., Hamadi, Y., Blake, A., 2006 "AutoCollage", ACM Transactions on Graphics (SIGGRAPH)

[15] Loui, A. C., Wood, M. D., 1999, "A software system for automatic albuming of consumer pictures", Proceedings of the 7th annual ACM Int. Conf. on Multimedia (Part 2), pp 159 - 162

[16] Tolstaya, E., "Automatic detection of digital image orientation", Russian patent application 2007102189, January 22, 2007

About the author

Ekaterina Tolstaya received her M. S. degree in applied mathematics from Moscow State University, in 2000. In 2004 she completed her M. S. degree in geophysics from University of Utah, USA. Her research in Utah involved inverse scattering in electromagnetics, and image processing for satellite sensing. Since 2004 she works on problems of image processing and reconstruction in Image Enhancement Technology Group, Printing Technology Lab, Samsung Research Center in Moscow, Russia.