AUTOMATIC IMAGE ORIENTATION DETECTION USING THE SUPERVISED SELF-ORGANIZING MAP

Manasi Datar[a] and Xiaojun Qi[b]
manasi.datar@ge.com and xqi@cc.usu.edu
[a] Imaging Technologies Lab, GE Global Research, Bangalore, India
[b] Computer Science Department, Utah State University, Logan, UT 84322-4205

ABSTRACT
Automatic detection and correction of image orientation is of great importance in intelligent image processing. In this paper, we present an automatic image orientation detection algorithm based on the supervised self-organizing map (SOM). The SOM is trained by using compact and efficient low-level chrominance (color) features in a supervised manner. Experiments have been conducted on a database containing various images with different compositions, locations, and contents. The proposed algorithm achieves an accuracy of 75% using the SOM trained by 600 images. In comparison with three peer systems, the proposed system achieves decent accuracy with the compact feature vector, the minimum training time, and the minimum training data. This framework will bridge the gap between computer and human vision systems and is applicable to other problems involving semantic content understanding.

KEY WORDS
Image orientation, and self-organizing map

1. Introduction

Important technological advances in digital imaging and networking have brought content-based image retrieval (CBIR) and organization to the forefront in computer vision and multimedia computing. In the meantime, development and easy availability of digital photography and storage equipment has made it convenient for digitizing and storing family and vacation photographs on personal computers. Consequently, there is an increasing demand for better image management systems that are capable of assisting the user in digitizing, storing, and retrieving images from digital libraries. All such systems require knowledge about the true image orientation, which is defined as "the orientation in which the scene, captured by the image, originally occurred" [1]. For instance, when a user scans a picture, she expects the resulting image to be displayed in its correct orientation, regardless of the orientation in which the photograph was placed on the scanner [2]. Fig. 1 illustrates such a scenario, where four possible orientations placed on the scanner are demonstrated.

Fig 1: Four possible orientations of an image. (The correct orientation detected by the proposed algorithm is 0°)

Research in automatic image orientation detection is still in its nascent stage since many CBIR systems assume that all images in the library are correctly oriented (i.e., in their true orientations). Some CBIR systems utilize other features such as composite region templates [3] to eliminate the needs for knowing the image orientation. Limited pioneer work on the image orientation detection is recently presented. In specific, a statistical learning support vector machine (SVM) [1] and a Bayesian learning framework [2] are respectively employed to classify the image orientations based on the low-level color features extracted from the image. Luo and Boutell [4] apply a probabilistic approach within a Bayesian framework to determine the image orientations by using a confidence-based integration of low-level and semantic cues.

In this paper, we present a novel framework for automatic image orientation detection. This framework is based on the supervised classification using the SOM. The peripheral regions of each image are used to extract the low-level regional color features in terms of the first and second color moments in the HSV color space. The constructed multi-dimensional feature vector is further used to represent each image in a supervised learning framework. The trained SOM is finally used for classifying the image orientations in terms of the four possible orientations shown in Fig. 1. The paper is organized as follows. Section 2 describes the various components in the proposed approach in detail. Section 3 reports the results. Section 4 draws conclusions and discusses future directions.

2. Proposed Approach

Our automatic image orientation detection system aims to determine the true orientation of an image from one of four possibilities of 0° (no rotation), 90°, 180° and 270°. Such an aim is valid since a deviation from true image orientation may result from either the camera rotation when capturing the extents of a certain scene or the placement of the image when scanning a picture. In spite of all possible rotation angles, it is safe to assume that the images are taken with either 0° or 90° by a camera or the images are placed on a scanner with their boundaries aligned with those of the scanner plate. The following subsections explain the proposed approach in detail.

2.1. Feature Extraction

We use low-level visual contents to represent an image. In specific, the image chrominance information, namely the 1st order (mean) and 2nd order (variance) color moments in the HSV color space, is extracted in our system since color moments are effective for color-based image analysis [5]. The 4 steps of the proposed feature extraction algorithm are:

1. Transform the image to the HSV color space.
2. Divide the image into $n \times n$ subblocks, where n is empirically set to be 4.
3. Calculate the 1st and 2nd order color moments for each peripheral block in each H, S, and V color plane.
4. These color features are reshaped into a feature vector to represent the image.

Fig. 2 illustrates our proposed subblocking scheme. The order to construct the feature vector using the color moments from the peripheral blocks is shown in Fig. 2(b). In our system, we exclusively use the features of the peripheral regions to represent an image since the peripheral regions of the image is more sensitive to orientations as compared to the central regions. Unlike other three systems [1, 2, 4], which extract both color moment and edge direction histograms to represent an image, we only extract color moments to represent an image. Such a choice is based on the observation that the color seems to be more essential for the orientation detection when using the peripheral regions. Consequently, the length of our feature vector for image orientation detection is 72 (i.e., 12*(3+3) = 72, where 12 is the number of peripheral blocks). Our feature vector has the shortest length compared with the feature vectors employed in other systems, whose lengths are 925 [1], 600 [2], and 719 [4], respectively. Such a compact image representation reduces time required to train the classifier.

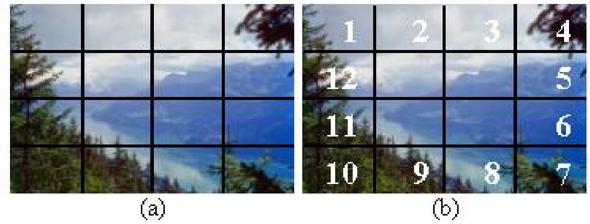

Fig. 2: Image subblocks:
(a) image grid with n = 4
(b) peripheral sub blocks used to compute color moments

2.2. A Supervised SOM-Based Classification

The SOM has been extensively used for unsupervised classification in the applications of image segmentation and pattern recognition. However, its application as a supervised classifier has only been investigated recently. In our system, a supervised SOM-based paradigm is employed for learning the image orientations, where a priori known class labels are used with the feature vectors in the training phase. This trained supervised SOM is further utilized for classifying the orientation of any unknown input image. The following sections explain the details of the steps involved in the classifier training and testing.

2.2.1 Training of the Supervised SOM

The supervised SOM [6] is a variant of the SOM. A SOM consists of m neurons located on a regular low-dimensional grid. The dimensionality of the grid is usually less than 3 due to problematic visualization of higher dimensions. The lattice of the grid is either hexagonal or rectangular. Each neuron i on the grid has a d-dimensional prototype vector denoted as $\mathbf{m}_i = [m_{i1} \ldots m_{id}]$. The training of a SOM is accomplished by iterating the following 4 steps until a topologically ordered mapping of the input data is created.

1. Randomly choose a sample data vector $\mathbf{x}$ from the training set.
2. Compute the distances between $\mathbf{x}$ and all prototype vectors located at m neurons.
3. Locate the best-matching unit (BMU) b, whose prototype vector is the closest to $\mathbf{x}$, by:
$$\|\mathbf{x}-\mathbf{m}_b\| = \min_i\{\|\mathbf{x}-\mathbf{m}_i\|\} \quad (1)$$
4. Move BMU b and its topological neighbors closer to the input vector in the input space by updating the their respective prototype vectors:
$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + \alpha(t)\,h_{bi}(t)[\mathbf{x}-\mathbf{m}_i(t)] \quad (2)$$
where t denotes time, $\alpha(t)$ is the learning rate, $h_{bi}(t)$ is a neighborhood kernel centered at b, and i is the index of the neighborhood units. In our system, this neighborhood kernel is chosen as a Gaussian kernel:

$$h_{bi}(t) = e^{-\frac{\|\mathbf{r}_b - \mathbf{r}_i\|^2}{2\sigma^2(t)}} \tag{3}$$

where $\mathbf{r}_b$ and $\mathbf{r}_i$ are the position vectors of b and i on the SOM grid, respectively.

During this training process, the SOM behaves like a flexible net that iteratively folds onto the “cloud” formed by the training data. In addition, the neighboring neurons are pulled towards the same direction as BMU b so their prototype vectors resemble each other. Fig. 3 graphically illustrates such updating process.

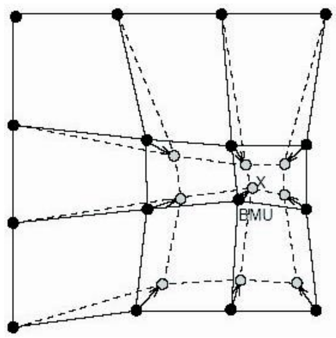

Fig. 3: SOM training: updating the BMU b and its neighbors to move them closer to input x.

For the supervised training, the label indicating the orientation of each training image is incorporated. The possible labels (i.e., classes) correspond to the counter-clockwise rotations with regards to North (0°), West (90°), South (180°), and East (270°). In our proposed system, we construct the training data by adding a 1-of-4 coded matrix to the original data based on their corresponding labels. As the training progresses, the label of each unit is determined by the maximum of the added labels. These extra components, 1-of-4 coded labels, are filtered out to represent a supervised SOM. Fig. 4 demonstrates a sample result of the supervised SOM, where each neuron is labeled with the orientation it represents. It clearly shows that the neurons with similar directions are clustered together, which is consistent with the step 4 in the training process where BMU and its topological neighbors are pulled towards the same direction.

2.2.2. Supervised SOM-Based Classification

Once the supervised SOM is trained, any test image can be sent in as an input and be classified to the appropriate label in the following manner:

1. Locate the BMU b using (1).
2. Extract the label corresponding to the BMU b from the trained SOM grid.

The extracted label indicates the current image orientation. If it indicates “north”, the image is in its true orientation. If it indicates “west”, a corresponding rotation (i.e., 90°) is then applied to bring the image back to its true orientation. The similar rule is applied to the other labels such as “south” and “east”. Fig. 5(a) shows the classification result of a sample image shown in Fig. 1(b). Fig. 5(b) demonstrates the restored image in its true orientation.

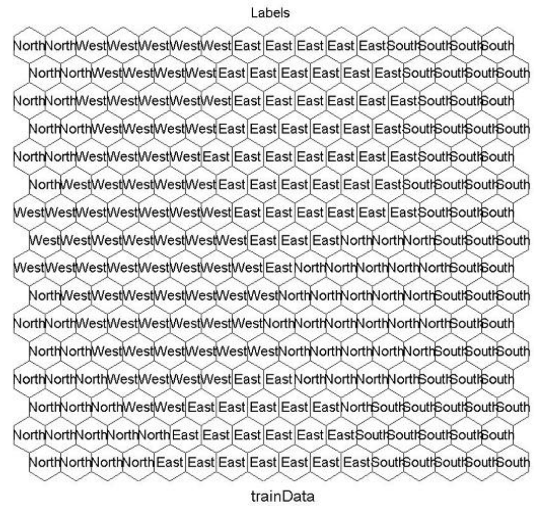

Fig. 4: Labeled SOM after the training

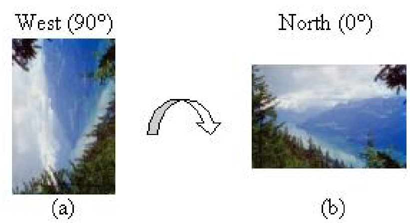

Fig. 5: Supervised SOM-based classification results

(a) detected orientation,

(b) image rotated to its correct orientation

3. Experimental Results

We have conducted extensive experiments to illustrate the performance of the proposed algorithm under four possible labels (i.e., classes), namely North (0° rotation), West (90° rotation), South (180° rotation), and East (270° rotation). Our database combines the images from the Corel photo library [4] with a collection of personal photographs to ensure enough variety of the data source.

The training set comprises 600 images from a subset of our database, which contain a variety of compositions (day/night illumination, etc.), locations (indoor/outdoor), and content (people, objects, etc.). The distribution of classes (i.e., image orientations) is kept equal. That is, we have 150 images in each of the four orientations.

The supervised SOM is initialized as a grid of 16×16 neurons arranged in a hexagonal lattice. The Gaussian kernel is set for the neighborhood and the initial prototype vectors are randomly initialized. Such a configuration is empirically tested to achieve the best orientation detection accuracy. Due to the compact feature vector, training the above configured SOM takes less than one minute in a supervised manner. The result of the training is the labeled SOM as shown in Fig. 4. A validation run is carried out on 100 randomly selected images from the training set. The algorithm achieves an accuracy of 97% on the training set.

An independent set of 600 images from another subset of our database is used as a testing set. The feature vector of each test image is extracted in the same manner as described in Section 2.1 and is then presented to the trained SOM for classification. The algorithm achieves an accuracy of 75% on the testing set. Some example image orientation results are shown in Fig. 6.

(a) (b)

Fig. 6. Subset of images with orientations correctly detected by the SOM.
(a) Input images
(b) Detected correct orientations

We have further studied the images whose orientations cannot be correctly detected by the proposed algorithm. We observe that these “troublesome” images are those with uniform texture or background, close-up images, and symmetric images as shown in Fig. 7. These images are difficult to detect their true orientations by using any current image orientation detection system. As a result, our proposed system should achieve better accuracy if such “troublesome” images are eliminated from the testing set.

Fig. 7. Examples for images whose orientations are not correctly detected by the proposed algorithm

Due to the unavailability of the common data sets and the executables, it is hard to compare our proposed system with the three peer systems [1, 2, 4] on the same basis. Table 1 simply compares our system with the three peer systems in terms of the feature vector length, the number of training images, and the orientation detection accuracy. It clearly shows that our proposed system achieves decent accuracy with the compact feature vector, the minimum training time, and the minimum training data.

Table 1: Comparisons of the proposed approach with three other approaches reported in literature

	SVM [1]	Bayesian [2]	Bayesian [4]	Supervised SOM
Feature Vector Length	925	600	719	72
Training Set Size	5416	7980	8316	600
Detection Accuracy	73%	98.6%	83%	75%

4. Discussions and Future Work

We have proposed an intuitive approach for automatic image orientation detection, based on supervised learning of the SOM. Our proposed feature extraction strategy considerably reduces the feature vector length as compared to the peers. The supervised SOM-based training is robust enough to work with a small training set and is capable of yielding comparable accuracies. The proposed system can be easily tuned to include any new arbitrary orientations and/or any new category of images.

Image orientation detection on consumer photographs is a new and little researched area. Hence, it should be noted that the algorithm compares favorably with the SVM approach [1], which includes such images in its experiments. The proposed algorithm achieves 75% accuracy in spite of using a very small training set. Training with a more comprehensive set of images will significantly improve the accuracy. Currently, the proposed algorithm only uses low-level visual cues as features. In the future, incorporation of high-level semantic cues will be considered to further improve the detection accuracy.

References

[1] Y.M. Wang & H. Zhang, Detecting Image Orientation Based on Low-Level Visual Content, *Computer Vision and Image Understanding*, 93, 2004, 328-346.

[2] A. Vailaya, H.-J. Zhang, C.-J. Yang, F.-I. Liu & A.K. Jain, Automatic Image Orientation Detection, *IEEE Transactions on Image Processing*, 11(7), 2002, 746-754.

[3] J. R. Smith & C.-S. Li, Image Classification and Querying Using Composite Region Templates, *Computer Vision and Image Understanding*, 75 (1-2), 1999, 165-174.

[4] J. Luo & M. Boutell, Automatic Image Orientation Detection via Confidence-Based Integration of Low-Level and Semantic Cues, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5), 2005, 715-726.

[5] W. Ma & H. Zhang, Content-Based Image Indexing and Retrieval, (Chapter13). In *The Handbook of Multimedia Computing*, B. Furht, ed. LLC, CRC Press, 1998.

[6] T. Kohonen, The Self-Organizing Map, in *Proceedings of the IEEE*, 78(9), 1990, 1464-1480.