MVA2005 IAPR Conference on Machine VIsion Applications, May 16-18, 2005 Tsukuba Science City, Japan
3-12

Shape-based Image Retrieval Using Support Vector Classification

Wai-Tak Wong
Department of Information Management
Chung Hua University
No. 707, Sec. 2, WuFu Road, HsinChu, Taiwan,
wtwong@mi.chu.edu.tw

Sheng-Hsun Hsu
Department of Information Management
Chung Hua University
No. 707, Sec. 2, WuFu Road, HsinChu, Taiwan,
spolo@mi.chu.edu.tw

Abstract

We have developed a novel method for shape-based image retrieval based on the Support Vector Machine technique (SVM) and the similarity measures. The high accuracy classification rate of SVM, 100% for 183 images in 8 categories from the public domain, shows that SVM is one of the best tools for classification problems. A sensitivity test is performed to show that SVM is quite robust against different parameter values. After the category of the input image is identified, our similarity measures are used to retrieve the similar shapes from the image database. Our method can satisfy necessary requirements of cognitively similarity measures from visual perception, such as rotation, scaling and shearing invariance.

1 Introduction

Shape-based retrieval of similar objects from image databases has been increasingly studied recently. Using similarity measures as indexes for image retrieval can be found in literatures [1]. However, seldom of them can be implemented in a large image databases. Shape similarity measures are essential in matching, which deals with transforming a pattern and measuring the degree of resemblance with another pattern [2]. Researchers are using different shape similarity measures to retrieve similar images [3]. Although these measures are effective, they don't particularly mention the conditions of affine transformation. However, image scaling or rotations are quite common situations because pictures are taken from different angles and with different sizes. To address this, we propose similarity measures based on the dominant points of the shape boundary. The advantage of these measures is that they are simple, efficient and invariant with respect to scaling, rotation and minor shearing.

In addition, we employ Support Vector Machine (SVM) to solve misclassification problem in the previous method [4]. SVM has recently been introduced as a new technique for solving a variety of learning, classification and prediction problems. Empirical testing has shown that the SVN performance is better than that of the ANN in classification problems [5]. The results show that our similarity measures are effective in that they help SVM to classify image objects. Finally, we found that SVM is quite robustness against different parameter values. This paper is organized as follows. Section 2 presents the dominant point detection. Section 3 introduces the roposed similarity measures. Section 4 briefly describes the SVM classification technique. Section 5 presents the experimental results followed by conclusions in Section 6.

2 Dominant point detection

The contour of an image region can be represented by chain codes – a starting point and a sequence of moves around the borders. Freeman's chain code is one of the most commonly used coding schemes. The direction of each movement is encoded by using a numbering scheme, such as $\{i|i=0,1,\ldots,7\}$ denoting an angle of $45i°$ counterclockwise from the positive x-axis. The chain codes therefore can be seen as a connected sequence of straight-line segments with specified lengths and directions. The chain code algorithm traces the border pixels one-by-one and generates codes by considering neighborhood pixels. However, an image might possess a large amount of chain code information. To effectively reduce the information, researchers usually employ the dominant point's (DP) concept. Dominant points are the points that hold the properties of having a high curvature on the contour of an image. Information on the shape of a curve is condensed at the dominant points. To date, most DP extraction algorithms consider a supporting region to minimize the number of points and errors [6]. However, these algorithms are both complicated and time-consuming. To address this, we develop a new method for generating dominant points based on corner points and Local Symmetry Deficiency (LSD) [7]. Our new method is not only simple and efficient, but also produces good quality of dominant points.

To effectively identify corner points, we define the total curvature variance TC_i at a point P_i with its chain code C_i as

$$TC_i = \sum_{j=-1}^{j=1} Mod(C_{i+j} - C_{i+j-1})$$

The function $Mod(C_{i+j} - C_{i+j-1})$ is defined as

$$\Delta C_{nm} \quad C_n - C_m$$

$$if\ ABS(\Delta C_{nm}) > 4\ then$$

$$\quad if\ \Delta C_{nm} > 0 \quad then \quad Mod(C_n \quad C_m) \quad \Delta C_{nm} - 8$$

$$\quad else \quad Mod(C_n \quad C_m) \quad \Delta C_{nm} + 8$$

$$else \quad Mod(C_n \quad C_m) = \Delta C_{nm}$$

After we calculate TC_i for every point on a curve, we compare it with the following criteria from:

(a) The total curvature variance $TC_i > 1$ (lie beyond 45°),

(b) ABS(TC_i)=1 (lie in 45°) and ($C_i \neq C_{i-1}$) and (($C_{i-1}=C_{i-2}$ AND $C_{i-2}=C_{i-3}$) or ($C_i=C_{i+1}$ AND $C_{i+1}=C_{i+2}$))

(c) TC_i=0 (look like a straight line) and ($C_i \neq C_{i-1}$) and (($C_{i-1}=C_{i-2}=C_{i-3}$ and $C_i=C_{i+2}$) or ($C_i=C_{i+1}=C_{i+2}$ and $C_{i-1}=C_{i-3}$))

(d) TC_i=0 (look like a straight line) and ($C_{i-1}=C_{i+1}$) and (($C_{i-1}=C_{i-2}=C_{i-3}=C_{i-4}$ and $C_{i+1}=C_{i+2}$) or ($C_{i+1}=C_{i+2}=C_{i+3}=C_{i+4}$ and $C_{i-1}=C_{i-2}$)).

A point P_i that satisfies one of the criteria is considered as a corner point. The obtained corner points are the candidates of dominant points. If there is no consecutive corner point, all candidates are chosen. If there are two consecutive corner points, we choose the one with the smallest LSD. If multiple consecutive corner points reside next to each other, three consecutive corner points are chosen as a group. We then choose the one with the smallest LSD to be the dominant point in the group. The rationale of LSD is to choose a dominant point with high symmetry at its vicinity.

3 The computation of the proposed similarity measures

Once we identify the dominant points, we can use them to calculate the similarity measures. First, we randomly pick one dominant point as a starting point. The lengths of chain codes between dominant points and the starting point are used as the perimeters for all dominant points; and the distances between the dominant points and the geometric center are considered as the geometric moments for all dominant points. We then normalize each perimeter and geometric moment to plot the "spectrum" of an image by the normalized geometric moment (*y*-axis) vs. the normalized perimeter (*x*-axis). The derived spectrum also holds very good property, as it is invariant with respect to image scaling, rotation, and minor shearing. For example, we show the spectrum of *Fish14* image from Sebastian's study [4,8] in Fig. 1. By the spectrum, we can obtain similarity measures for SVM. The first similarity measure is the total normalized area (*TNA*) – areas covered under the spectrum. The second measure is the total perimeter of the spectrum, which is the total normalized moment variance (*TNMV*). The computations are shown as follows: where d denotes the number of dominant points, M denotes the moment at point i, ΔM denotes difference of the moment between two consecutive dominant points, NP denotes the normalized perimeter length, and ΔD denotes the normalized perimeter difference between two dominant points.

$$TNA = \sum_{i=0}^{i=d} \frac{(M_i + M_{i+1}) * \Delta D}{2} \quad where \quad \Delta D = NP_{i+1} - NP_i$$

$$TNMV = \sum_{i=0}^{i=d} \sqrt{(\Delta M)^2 + (\Delta D)^2} \quad where \quad \Delta M = M_{i+1} - M_i$$

Other measures are also computed as the input features for SVM, including the cross-sectional normalized area (*TCSNA*) and the cross-sectional normalized moment variance (*TCSNMV*). We derive these measures by dividing the *TNA* and *TNMV* according to the five equal partitions of the *y-axis*. Then we obtain 5 sets of *TCSNA* and *TCSNMV*. In total, we have 12 attributes as the input features for SVM.

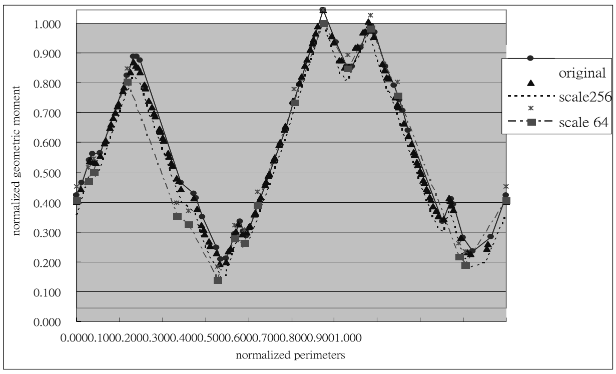

Figure 1a. The spectrums of the Fish 14, after half size and double size scaling.

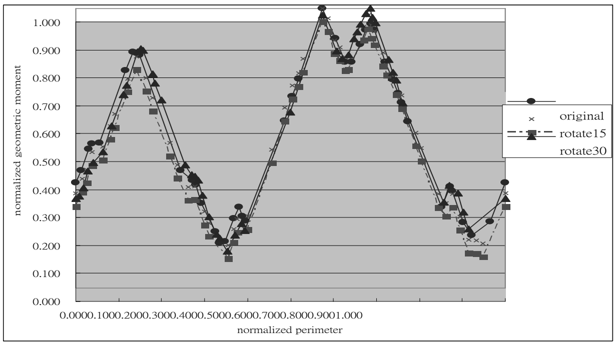

Figure 1b. The spectrums of the Fish 14, after 30° and after 45° rotation.

4 Support Vector Machine

SVM originated as an implementation of Vapnik's [9] structural risk minimization (SRM) principle, which minimizes the generalization error, i.e. true error on unseen examples. The basic idea of SVM is to transform data into a higher dimensional feature and find the optimal hyper plane in the space that maximizes the margin between classes. The simplest SVM deals with a two-class problem - in which the data is separated by a hyper plane defined by a number of *support vectors*. Support vectors are a subset of training data used to define the boundary between the two classes. In situations where SVM cannot separate two classes, it solves this problem by mapping input data into high-dimensional feature spaces using a kernel function. In high-dimensional space it is possible to create a hyper plane that allows linear separation – corresponding to a curved surface in the lower-dimensional input space. Accordingly, the kernel function plays an important role in SVM. In practice, various kernel functions can be used, such as linear, polynomial or Gaussian.

In the two-dimensional case, the SVM attempts to place a linear boundary between the two different classes and orients this line in such a way that the margin is maximized. The nearest data points used to define the margin are known as support vectors. Support vectors, not the number of input features, contain all of the information needed to define the classifier. One remarkable property of SVM is its ability to learn can be independent of the feature space dimensionality. This means that SVM can generalize well in the presence of many features. The simplest model of SVM is called the maximal margin classifier. It works only for data that are linearly separable in the feature space. Though it is the easiest algorithm and not very useful in real-world situations, it forms the building block for understanding the complex SVM models.

Mathematically, the linear boundary can be expressed in terms of

$$w^T x + b = 0$$

In a classification problem, we try to estimate a function

$$f: R^n \mapsto \{\pm 1\}$$

using training data. Let us denote the class A with $x \in A$, $y = 1$ and class B with $x \in B$, $y = -1$; $(x_i, y_i) \in R^n \times \{\pm 1\}$. If the training data are linearly separable then there exists a pair (w, b) $\in R^n \times R$ such that

$$w^T x + b \geq 1, \text{ for all } x \in A$$
$$w^T x + b \leq -1, \text{ for all } x \in B$$

where w is termed the weight vector and b the bias. The inequality constraints can be combined to give

$$y(w^T x + b) \geq 1, \text{ for all } x \in A \cup B$$

The maximal margin classifier optimizes this by separating the data with the maximal margin hyper plane. Instead of maximizing $2/\|w\|$, we minimize $\|w\|^2/2$ The learning problem is hence formulated as: minimize $\|w\|^2/2$ subject to the constraints of linear separability. The optimization problem in essence is a quadratic programming problem:

$$\underset{\mathrm{w},b}{\text{Minimize}}\ \Phi(\mathrm{w}) = \frac{1}{2}\|\mathrm{w}\|^2 \qquad s.t.\ y(\mathrm{w}^T\mathrm{x} + b) \geq 1$$

This problem has a global optimum solution. By applying the Lagrange relaxation method, the problem can be formulated as follows:

$$L_P(\mathrm{w}, b, \Lambda) = \frac{1}{2}\|\mathrm{w}\|^2 - \sum_{i=1}^{l} \lambda_i [y_i(\mathrm{w}^T\mathrm{x}_i + b) - 1],$$

where $\Lambda = \{\lambda_1, \cdots \lambda_l\}$ are the Lagrange multipliers. The Lagrangian L has to be minimized with respect to the primal variables w and b. Differentiating with respect to w and b and setting the derivatives equal to 0 yields

$$\frac{\partial L_P(\mathrm{w}, b, \Lambda)}{\partial \mathrm{w}} = \mathrm{w} - \sum_{i=1}^{l} \lambda_i y_i \mathrm{x}_i = 0$$

and

$$\frac{\partial L_P(\mathrm{w}, b, \Lambda)}{\partial \mathrm{b}} = -\sum_{i=1}^{l} \lambda_i y_i = 0$$

Substituting into the equation we obtain the dual form of the optimization problem:

$$\underset{\lambda_i}{\text{Maximize}}\ L_D = \sum_{i=1}^{l} \lambda_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} \lambda_i \lambda_j y_i y_j \mathrm{x}_i^T \mathrm{x}_j,$$

The first term of the above equation shows that the solution vector has an expansion in terms of a subset of the training patterns, namely those patterns whose Lagrange multiplier is non-zero. The optimization problem can be determined by some training points (support vectors) and λ_i. Kuhn and Tucker extended the Lagrangian theory by incorporating with inequality constraint properties, providing a necessary and sufficient condition for the optimal solution. By the Karush-Kuhn-Tuker (KKT) complementary conditions, we obtain

$$\lambda_i [y_i(\mathrm{w}^T\mathrm{x}_i + b) - 1] = 0, \qquad i = 1, \ldots, l,$$

Note that the Lagrange multipliers are only non-zero when $y(wx+b)=1$ and the vectors satisfied such equation are called support vectors because they lie closest to the separating hyper plane. Consequently, the decision function is determined by a small subset of training set; other points are irrelevant in terms of decision function. With support vectors, over fitting is unlikely to occur [10]. The decision function can be reduced to contain only support vectors

$$f(\mathrm{x}) = \mathrm{sign}\left(\sum_{i=1}^{l} y_i \lambda_i^* (\mathrm{x}^T\mathrm{x}_i) + b^*\right)$$
$$= \mathrm{sign}\left(\sum_{i \in \text{support vectors}} y_i \lambda_i^* (\mathrm{x}^T\mathrm{x}_i) + b^*\right)$$

However, the real-world situations are not so perfect, they are seldom linearly separable. In order to cope with this issue, non-negative slack variables and kernel functions are introduced to deal with non-linear decision surfaces. Slack variables, ξ, are incorporated into constraints as follows:

$$L(\mathrm{w}, b) = \frac{1}{2}\|\mathrm{w}\|^2 + C\sum_{i=1}^{l} \xi_i$$

subject to

$$y(\mathrm{w}^T\mathrm{x} + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \text{ for } i = 1, \ldots, l,$$

where C is chosen by users. C is a regularization parameter that controls the trade-off between maximizing the margin and minimizing the training error.

5 Experimental Results

The data used here was originated from Sebastian's study [4,8]. To test our similarity measures, we randomly chose 8 categories with 183 pictures as shown in Fig. 2. All training images were classified to the right categories in the image database. Each image was associated with the attributes, *TNA, TNMV, TCSNA* and *TCSNMV*. For a testing image, it was input to the SVM to identify the correct category first. Then the similar images were retrieved from the image database according to the category and a given tolerance of the similarity measures.

A SVM implementation called LIBSVM[1] was used in this work. In our experiment, we chose the Gaussian kernel $k(x,y)=exp(-(x-y)^2)/\delta^2$ as our kernel function because it tends to achieve better performance. To determine the kernel bandwidth δ^2 and the margin *C*, ten-fold cross validation was used in the training data set to choose parameters that yield the best result. The parameters were $\delta^2 \in \{2,1,0.1,0.01,0.001,0.0001\}$ and $C \in \{1000,750,500,100,50,2\}$. Subsequently, this set of parameters was applied to the test data sets. The parameters chosen were δ^2 = 1and C = 1000. The accuracy of prediction rate on testing data for SVM is 100%.

To understand the effects of changing two important parameters (kernel parameters δ^2 and margin *C*) on the quality of image retrieval, we have run 20 experiments as shown in Table 1. We have changed δ^2 from 5 to 2, 1 and 0.5, while changing *C* from 1 to 10, 50, 100 and 1000, respectively. The quality of the image retrieval is measured by the prediction accuracy on both training and testing dataset. We observed that the parameter combinations yield quite stable results when *C* reached 10, (both the results on training and testing set > 95%). This suggests that SVM is quite robust against parameter selections.

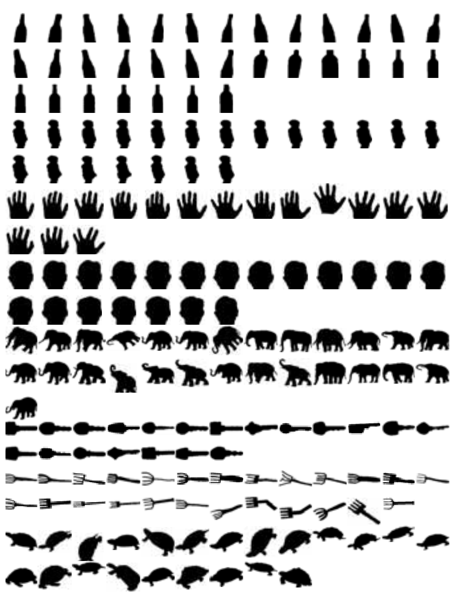

Figure 2. 183 shapes in 8 categories from Sebastian[4,8].

6 Conclusions

As the issue of image retrieval becomes more important, researchers must keep on developing effective methods to retrieve images. A set of similarity measures that is invariant with respect to image rotation, scaling and minor shearing is proposed in this paper. Our preliminary experiments with the Sebastian's data showed that these measures are not only invariant, but also quite effective in that they help SVM to achieve high (100%) classification rates. The image retrieval process is speeded up by indexing the proposed similarity measures on the image database.

Table 1. Sensitivity of SVM to parameters

$C\backslash\delta^2$	5		2		1		0.5	
	Train.	Test	Train.	Test	Train.	Test	Train.	Test
1	97.1	95.5	95.7	95.5	92.8	95.5	91.3	86.4
10	98.6	95.5	98.6	95.5	97.1	97.7	97.1	95.5
50	99.3	95.5	99.3	95.5	98.6	97.7	97.8	95.5
100	100	97.7	99.3	95.5	98.6	97.7	98.6	95.5
1000	100	97.7	100	97.7	100	100	99.3	97.7
Avg.	99	96.4	98.6	96	97.4	97.8	96.8	94.1

Acknowledgements

This work is supported by National Science Council, Taiwan, ROC under the grant NSC 91-2218-E-216-001.

References

[1] Editorial, "Special issue: shape representation and similarity for image databases,'' *Pattern Recognition*, vol. 35, pp. 1-2, 2002.

[2] R.C. Veltkamp, "Shape matching: similarity measure and algorithms," *Proceedings of International Conference on Shape Modeling and Applications (SMI'01),* May 2001, pp. 188 – 197.

[3] S. Santini, R. Jain, "Similarity measures," *IEEE Trans. Pattern Anal. and Mach. Intell.* vol. 21, no. 9, pp. 871-882, 1999.

[4] T.B. Sebastian, P.N. Klein, B.B Kimia, "Shock-based indexing into large shape databases," *Proceedings of Europe Conf. on Computer Vision,* Copenhagen, Denmark, May 2002, pp. 731-746.

[5] Y.D. Cai, X.J. Lin, X.B. Xu, K.C. Chou, "Prediction of protein structural classes by support vector machines," *Computers and Chemistry* vol. 26, pp. 293-296. 2002.

[6] R. Neumann, G. Teisseron, "Extraction of dominant points by estimation of the contour fluctuactions," *Pattern Recognition*, vol. 35, pp. 1447-1462, 2002.

[7] F.Y. Shih, W.-T. Wong, "A one-pass algorithm for local symmetry of contours from chain codes," *Pattern Recognition*, vol. 32, pp. 1203-1210, 1999.

[8] T.B. Sebastian, P.N. Klein, B.B Kimia, "Recognition of shapes by editing their shock graphs," *IEEE Trans. Pattern Anal. and Mach. Intell.,* vol. 26, no. 5, pp. 550-571, 2004.

[9] V.N. Vapnik, The Nature of Statistical Learning Theory. Springer-Verlag: New York, 1995.

[10] Witten I.H., Frank E., Data mining: practical machine learning tools and techniques with java implementations. Academic Press: US, 1999.

[1] http://www.csie.ntu.edu.tw/~cjlin/libsvm/