Image Retrieval Systems Based On Compact Shape Descriptor and Relevance Feedback Information

Konstantinos Zagoris\textsuperscript{a}, Kavallieratou Ergina\textsuperscript{b}, Nikos Papamarkos\textsuperscript{a}

\textsuperscript{a}Department of Electrical \& Computer Engineering Democritus University of Thrace, 67100 Xanthi, Greece
\textsuperscript{b}Department of Information and Communication Systems Engineering University of the Aegean Samos 83100, Greece

Abstract

One of the most important and most used low-level image feature is the shape employed in a variety of systems such as document image retrieval through word spotting. In this paper an MPEG-like descriptor is proposed that contains conventional contour and region shape features with a wide applicability from any arbitrary shape to document retrieval through word spotting. Its size and storage requirements are kept to minimum without limiting its discriminating ability. In addition to that, a Relevance Feedback technique based on Support Vector Machines is provided that employs the proposed descriptor with the purpose to measure how well it performs with it. In order to evaluate the proposed descriptor it is compared against different descriptors at the MPEG-7 CE1 Set B database.

Key words: Retrieval System, Document Retrieval, Handwritten Words, Word Spotting, MPEG-7, Relevance Feedback, Support Vector Machines, Shape

1. Introduction

Shape is one of the most important low-level image feature employed in content based retrieval systems. It can be used in a variety of ways. One such way is the document image retrieval through word spotting. Word spotting is a term borrowed by speech recognition field and introduced in document image retrieval by Manmatha [1]. It includes the localisation of parts of text specified by query words, comparing a user provided template of text image. It is a generic approach that can be applied to any document
written in any language using any alphabet, pictograph or ideogram.

Bhardwaj et al. [2] describe a method for script independent word spotting in multilingual handwritten and machine printed documents. The system accepts a query in the form of text from the user and returns a ranked list of word images from document image corpus based on similarity with the query word. The system is divided into two main components. The first component known as Indexer, performs indexing of all word images present in the document image corpus. This is achieved by extracting Moment Based features from word images and storing them as index. A template is generated for keyword spotting which stores the mapping of a keyword string to its corresponding word image which is used for generating query feature vector. The second component, Similarity Matcher, returns a ranked list of word images which are most similar to the query based on a cosine similarity metric. A manual Relevance feedback is applied based on Rocchio’s formula, which re-formulates the query vector to return an improved ranked listing of word images.

Adamek et al. [3] proposed a new approach to holistic word recognition for historical handwritten manuscripts based on matching word contours instead of whole images or word profiles. The new method consists of robust extraction of closed word contours and the application of an elastic contour matching technique proposed originally for general shapes [2]. They demonstrate that contour-based descriptors can effectively capture intrinsic word features.

The Moving Picture Experts Group (MPEG) [4, 5, 6] expresses a set of principles that the descriptors must maintain such as good retrieval accuracy, general application and compact form.

In this paper a MPEG-like descriptor is proposed that contains conventional contour and region shape features with a wide applicability from any arbitrary shape to document retrieval through word spotting. The proposed descriptor is compact (it less than 16 bytes) and requires low computational cost. The obtained retrieval results are satisfactory in comparison with other MPEG-7 shapes descriptors. Particular, it has been designed with attention to its size and storage requirements, keeping it as small as possible without limiting its discriminating ability. Also, a Relevance Feedback technique is provided that employs the above descriptor with the purpose to measure how well it performs with it.

In order to evaluate the proposed descriptor it is compared against various and different descriptors. Also, to test its general applicability three different databases are employed: A database that contains Greek handwritings

2. Compact Shape Portrayal Descriptor (CSPD)

2.1. Overview

The Compact Shape Portrayal Descriptor is a forty one dimension vector that it is created from five distinct features that capture satisfactory the shape of an object. Also, each vector value is quantized for binary representation in three bits so the storage requirements are only 123 bits per shape. As the block diagram portrays (Figure 1), each feature can be calculated separately so the computation of the CSPD can be also, easily parallellized.

2.2. Features Extraction

The proposed descriptor is based on three powerful features that are extracted from every shape (or word) and they are capable of capturing its similarities and discarding the small differences due to remained noise or different style of fonts. The three-feature-set is:

*Width to Height Ratio*: It is calculated as:

\[
WHR = \frac{\min\{W, H\}}{\max\{W, H\}}
\]  

*Vertical - Horizontal Projection*: This feature consists of a twenty dimension vector created from the quantified normalized coefficients of the Discrete Cosine Transform (DCT) of the normalized and smoothed vertical and horizontal shape projections (Figure 2). Its calculation consists of the following steps:

**Step 1**: From the following equation (Eq. 2) a new normalized vertical projection \( VP[i] \) is produced which has maximum height equal to one

\[
VP[i] = \frac{VP_{orig}[i]}{\max\{VP_{orig}\}}, \ i \in [1, W]
\]  

where \( VP_{orig}[i] \) is the original vertical projection of the shape, \( \max\{VP_{orig}\} \) the maximum value of \( VP_{orig} \) and \( W \) is the Width of the shape. The same is applied to the horizontal projection \( HP[i] \), too:

\[
HP[i] = \frac{HP_{orig}[i]}{\max\{HP_{orig}\}}, \ i \in [1, H]
\]
Figure 1: The block diagram of the CSPD calculation.
Figure 2: A visual representation of the Vertical - Horizontal Projection calculation for the word image "Returns" and for the shape of camel. (a) Original word image. (b) The original Vertical Projection of the word. (c) The smoothed and normalized Vertical Projection. (d) The original Horizontal Projection of the word. (e) The smoothed and normalized Horizontal Projection. (f) The shape of a camel. (g) The original Vertical Projection of the shape. (h) The smoothed and normalized Vertical Projection. (i) The original Horizontal Projection of the shape. (j) The smoothed and normalized Horizontal Projection.
where $HP_{\text{orig}}[i]$ is the original horizontal projection of the shape, $\max\{HP_{\text{orig}}\}$ the maximum value of $HP_{\text{orig}}$ and $H$ is the Height of the shape.

**Step 2:** The final normalized vertical and horizontal projections, depicted in the Figures 2(c) and 2(h) are created after a $3 \times 1$ mean mask is applied to the two projections $VP[i]$ and $HP[i]$. This way, the final projections are more robust to the changes of the noise or font size and type, as the small differences are smoothed.

**Step 3:** Then a Discrete Cosine Transform (DCT) is applied to the two normalized projections.

**Step 4:** The next step consists of the normalization of the coefficients of the Vertical and Horizontal DCT based on the following equations:

$$VP_{\text{NDCT}}[i] = \frac{VP_{\text{DCT}}[i + 1]}{VP_{\text{DCT}}[0]}, \quad i \in [1, 10]$$  (4)

where $VP_{\text{NDCT}}$ are the normalized coefficients and $VP_{\text{DCT}}$ the original coefficients of the $VP[i]$.

$$HP_{\text{NDCT}}[i] = \frac{HP_{\text{DCT}}[i + 1]}{HP_{\text{DCT}}[0]}, \quad i \in [1, 10]$$  (5)

where $HP_{\text{NDCT}}$ are the normalized coefficients and $HP_{\text{DCT}}$ the original coefficients of the $VP[i]$.

This step is necessary in order to eliminate the influence of the shape width and height in the vertical and horizontal projections, respectively. Worth noticing, that the $HP_{\text{NDCT}}$ and $VP_{\text{NDCT}}$ are 10-dimension vectors so only the first 11 DCT coefficients must calculated. The quantization of the normalize DCT coefficients is described at section 2.3. Also, the numbers of the required coefficients decided after extensive evaluations (See Section 5.4).

**Top - Bottom Shape Projections:** As it is shown in Figure 3, the Top-Bottom Shape Projections can be considered as signatures of the word shape. These signatures lead to a 20-dimension feature vector, where the first 10 values are calculated from the quantified normalized coefficients of the smoothed and normalized Top Shape Projection DCT (Figure 3(c) and 3(h)) and the rest 10 values from the quantified normalized coefficients of the smoothed
and normalized Bottom Shape Projection DCT (Figure 3(e) and 3(j)).
In order to calculate the Top Shape Projection, the word image is scanned from top to bottom. As it is shown in Figure 3(b) and (g), the first time a black pixel is found all the following pixels of the same column are converted to black.

The Bottom Shape Projection is calculated similarly. As it is depicted in Figure 5(d) and 5(i), the word image is scanned from bottom to top and all the pixels are converted to black until a black pixel is found. The rest of the processing steps of the Top and Bottom Shape Projections are the same of those of Vertical and Top Projections.

Figure 3: A visual representation of the Top-Bottom Shape Projection calculations for the word image "Returns" and for the shape of camel. (a) Original word image. (b) The original Top Shape Projection of the word. (c) The smoothed and normalized Top Shape Projection. (d) The original Bottom Shape Projection of the word (e) The smoothed and normalized Bottom Shape Projection. (f) The shape of a camel (g) The original Top Shape Projection of the shape. (h) The smoothed and normalized Top Shape Projection. (i) The original Bottom Shape Projection of the shape (j) The smoothed and normalized Bottom Shape Projection.
2.3. Quantization

In order to compress the descriptor even more, the values of the feature vectors are quantized for binary representation in three bits for each element of the descriptor. So the storage requirement is equal to $3 \times 41 = 123$ bits. The values of the descriptor are concentrated within small ranges so they must be non-linearly quantized in order to minimize the overall number of bits. Also, each feature is not related to each other so they must have differing quantization values.

The quantization is achieved by the Gustafson-Kessel (GK) [9] fuzzy algorithm which it is an extension of the fuzzy C-mean algorithm (FCM). The advance of the GK algorithm is that produces ellipsoidal clusters by using a covariance matrix instead the spherical clusters that produce many other clustering algorithms (FCM, Kohonen Self Organized Featured Map, etc) as Figure 4 illustrates. The act of quantization by the GK algorithm is employed successfully in others compact composite descriptors by content-based retrieval systems of natural images [10, 11, 12]. Also, the MPEG-7 quantizes its compact descriptors, too.

In Gustafson-Kessel algorithm, each cluster is characterized by its center

$$d_{ik}^2 = (x_k - v_i)^T A_i (x_k - v_i)$$

\[ k \in [1,n] \text{ and } i \in [1,c] \]

Figure 4: The ability of Gustafson-Kessel algorithm to produce ellipsoidal clusters: (a) The points in the 2D space to cluster in 4 groups. (b) Four clusters obtained by the fuzzy c-means algorithm. (c) The four clusters obtained by the Gustafson - Kessel algorithm.
The topological structure of the data inside a class and it is calculated from the following equation:

\[ A_i = \left( \rho_i \det (F_i) \right)^{\frac{1}{h}} F_i^{-1}, \quad i \in [1, c] \]  

(7)

where \( h \) represent the number of dimensions of the space that the data reside. Because the feature space has one dimension, the value of \( h \) is equal to 1 in this work. \( F_i \) is the covariance matrix which shows how the samples are scattered inside the cluster:

\[ F_i = \frac{\sum_{k=1}^{n} (u_{ik})^m (x_k - v_i)(x_k - v_i)^T}{\sum_{k=1}^{n} (u_{ik})^m}, \quad i \in [1, c] \]  

(8)

The weighting parameter \( m, m \in (1, \infty) \) influences the crispness or the fuzziness of the partition between the clusters. In our work \( m = 2 \). Worth noticing that if the Eq. 7 and 8 substituted into the Eq. 6, the outcome will be a squared Mahalanobis distance norm.

Finally, the \( U = [u_{ik}] \) is called partition matrix and it is defined as the grade of membership of \( x_k \) to the cluster \( i \) and it must satisfy the following constraints:

\[ 0 \leq u_{ik} \leq 1, \quad i \in [1, c] \quad \text{and} \quad k \in [1, n] \]  

(9)

\[ \sum_{i=1}^{c} u_{ik} = 1, \quad k \in [1, n] \]  

(10)

\[ 0 < \sum_{k=1}^{n} u_{ik} < n, \quad i \in [1, c] \]  

(11)

In order to calculate the CSPD quantization table (Table 1), a collection is assembled randomly from the shapes and words that reside in the evaluation databases. The size of the sample collection is chosen arbitrary to 1500 as the differences that GK algorithm produces for largest collection is meaningful.

From them, sample sets are created which they are corresponding to each feature vector. So from the collection 1500 (1500x1) samples for the Weight to Height Ratio, 15000 (1500x10) for the Vertical Projection, 15000 (1500x10) for the Horizontal Projection, 15000 (1500x10) for the Top Shape Projection and 15000 (1500x10) for the Bottom Shape Projection are extracted.
Then the above samples are constituted inputs into the GK classifier, which separates the volumes of the samples in regions, mapping each value from the decimal area to the integer area $[0,7]$ or to the binary area $[000, 111]$. The Gustafson-Kessel algorithm produces eight clusters which they are defined by a center $v$ and a positive-define matrix $A$ adapted according to the topological structure of the data inside the cluster. Each decimal value is mapped to the smallest integer/binary value distance according to Eq. 6. The outputs of the quantization are assembled in a 41-bin vector, which the 1st bin is reserved for the Width to Height Ratio, the 2nd - 11th bin for the Vertical Projection, the 12th-21th bin for the Horizontal Projection, the 22th-31th for the Top Shape Projection and 31th-41th for the Bottom Shape Projection as Figure 1 depicts. Also, the schema of the CSPD descriptor as an MPEG-7 extension of its visual descriptors schema is presented in Listing 1.

Table 1: CSPD Quantization Table.

<table>
<thead>
<tr>
<th>Width to Height (1st bin)</th>
<th>Value</th>
<th>000 = 0</th>
<th>001 = 1</th>
<th>010 = 2</th>
<th>011 = 3</th>
<th>100 = 4</th>
<th>101 = 5</th>
<th>110 = 6</th>
<th>111 = 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v$</td>
<td>0.194</td>
<td>0.358</td>
<td>0.478</td>
<td>0.606</td>
<td>0.733</td>
<td>0.815</td>
<td>0.89</td>
<td>0.975</td>
<td></td>
</tr>
<tr>
<td>$A$</td>
<td>25.013</td>
<td>26.669</td>
<td>30.004</td>
<td>31.839</td>
<td>35.785</td>
<td>52.873</td>
<td>47.896</td>
<td>58.456</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Vertical Projection (2nd - 11th bin)</th>
<th>Value</th>
<th>000 = 0</th>
<th>001 = 1</th>
<th>010 = 2</th>
<th>011 = 3</th>
<th>100 = 4</th>
<th>101 = 5</th>
<th>110 = 6</th>
<th>111 = 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v$</td>
<td>0.504</td>
<td>-0.215</td>
<td>-0.09</td>
<td>-0.031</td>
<td>0.004</td>
<td>0.069</td>
<td>0.392</td>
<td>1.412</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Horizontal Projection (12th - 21th bin)</th>
<th>Value</th>
<th>000 = 0</th>
<th>001 = 1</th>
<th>010 = 2</th>
<th>011 = 3</th>
<th>100 = 4</th>
<th>101 = 5</th>
<th>110 = 6</th>
<th>111 = 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v$</td>
<td>-0.456</td>
<td>-0.194</td>
<td>-0.09</td>
<td>-0.035</td>
<td>0.003</td>
<td>0.064</td>
<td>0.252</td>
<td>1.414</td>
<td></td>
</tr>
<tr>
<td>$A$</td>
<td>9.781</td>
<td>21.917</td>
<td>47.372</td>
<td>76.465</td>
<td>90.791</td>
<td>38.51</td>
<td>9.305</td>
<td>69.845</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Top Shape Projection (22th - 31th bin)</th>
<th>Value</th>
<th>000 = 0</th>
<th>001 = 1</th>
<th>010 = 2</th>
<th>011 = 3</th>
<th>100 = 4</th>
<th>101 = 5</th>
<th>110 = 6</th>
<th>111 = 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v$</td>
<td>-0.438</td>
<td>-0.126</td>
<td>-0.049</td>
<td>-0.014</td>
<td>0.003</td>
<td>0.054</td>
<td>0.505</td>
<td>1.413</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bottom Shape Projection (32th - 41th bin)</th>
<th>Value</th>
<th>000 = 0</th>
<th>001 = 1</th>
<th>010 = 2</th>
<th>011 = 3</th>
<th>100 = 4</th>
<th>101 = 5</th>
<th>110 = 6</th>
<th>111 = 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v$</td>
<td>-0.677</td>
<td>-0.136</td>
<td>-0.003</td>
<td>0.079</td>
<td>0.206</td>
<td>0.436</td>
<td>0.885</td>
<td>1.413</td>
<td></td>
</tr>
</tbody>
</table>
Listing 1: The schema of the CSPD descriptor as an extension of the MPEG-7 Visual Descriptors

```xml
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
 xmlns:mpeg7="urn:mpeg:mpeg7:schema:2004"
 xmlns:CSPDNS="CSPDNS" targetNamespace="CSPDNS"
 import namespace="urn:mpeg:mpeg7:schema:2004"
 schemaLocation="Mpeg7-2004.xsd" />
<complexType name="CSPDType" final="#all">
 <complexContent>
  <extension base="mpeg7:VisualDType">
   <sequence>
    <element name="value">
     <simpleType>
      <restriction>
       <simpleType>
        <list itemType="mpeg7:unsigned3" />
       </simpleType>
       <length value="41" />
      </restriction>
     </simpleType>
    </element>
   </sequence>
   <extension base="mpeg7:VisualDType">
   </extension>
  </complexContent>
 </complexType>
</schema>
```
3. Relevance Feedback Information

3.1. Introduction

By adopting relevance feedback algorithms, high retrieval scores can be attained in retrieval systems. These algorithms require the user to grade the quality of the query results by tagging the retrieved images as being either correct or not. Then, the search engine uses this information in subsequent queries to better satisfy users’ needs. Also, sometimes there is not an appropriate query image to use for retrieval.

The proposed Relevance Feedback (RF) algorithm attempts to overcome these problems by providing a mechanism to fine tune the retrieval results. This is achieved by letting the user to define a number of retrieved objects as correct or wrong. Then the original query descriptor and those from the images that the user tagged as being either correct or not are used as training samples to the Support Vector Machines (SVMs) [13, 14]. Final, new retrieval results are calculated by using the output of decision function of the trained SVMs as similarity measure.

We implement an RF algorithm in order to test how the proposed descriptor performs in those systems. Also, we propose a new algorithm for detecting the SVMs parameters inspired from the Parameter Estimation Algorithm [15] for the binarization methods.

3.2. Support Vector Machines

The Support Vector Machines (SVMs), introduced in 1992 [13, 14] are based on statistical learning theory and recently have been applied to many and various classification problems. 

\[ D \] is defined as a given training dataset \( \{(x_i, y_i)\}_{i=1}^{n}, x \in [0, 1], y \in \{-1, +1\}, i \in [1, n] \), where \( x_i \) is the \( i \)th input vector and \( y \) is the label correspond to the \( x_i \).

If the training data are not linear separable then they mapped from the input space to a feature space using the kernel method, defined as:

\[ k(x, x') = \varphi(x)^T \varphi(x') \]  

where \( \varphi(x) \) is the feature map, which it is mapping the input space to a high dimensional feature space where the training data become linearly separable.

Taking account the structure of the proposed descriptor the Radial Basis
Function \((\exp\{-\gamma \|x - x'\|\})\) is used as a kernel.

So, the SVMs classifier satisfy the following conditions:

\[ y_i \left[ \alpha_i k(x, x_i) + b \right] - 1 + \xi_i \geq 0 \quad (13) \]

The \(\xi_i > 0\) are slack variables and they are used for misclassifying some data points (for instance to overcome the over fitting problem).

Finally, the maximum margin classifier is calculated by solving the following constrained optimization problem which it is expressed in terms of variables \(\alpha_i\):

\[
\begin{align*}
\max_{\alpha} & \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j x_i^T x_j \\
\text{subject to:} & \quad \sum_{i=1}^{n} y_j \alpha_i = 0, \quad 0 \leq \alpha_i \leq C 
\end{align*}
\quad (14)
\]

The constant \(C > 0\) defines the trade-off between the training error and the margin. The training data \(x_i\) for which \(\alpha_i > 0\), are called support vectors.

In SVMs, the function that decides the classification of the data \(x\) to the two classes is:

\[ f(x) = \text{sign} \left( \sum_{i=1}^{n} a_i y_i (\phi(x_i), \phi(x)) - b \right) \quad (15) \]

So if \(f(x) > 0\) the data \(x\) is classified to the class 1 otherwise it is classified to the class 0. We proposed the utilization of the decision function value instead its sign to calculate the membership function of the data \(x\) to the class 1 according to the Eq. 16:

\[
R(x) = \begin{cases} 
100 \times \max \left\{ \frac{1}{1 + \frac{1}{3}e^{f(x)}}, \frac{1}{1 + \frac{1}{3}e^{-f(x)}} \right\} & \text{if } f(x) > 0 \\
100 \times \left( 1 - \max \left\{ \frac{1}{1 + \frac{1}{3}e^{f(x)}}, \frac{1}{1 + \frac{1}{3}e^{-f(x)}} \right\} \right) & \text{if } f(x) < 0 
\end{cases} \quad (16)
\]

The output of the Eq. 16 is a number between 0 and 100, which represents the membership value of the data \(x\) to the class 1.

One of the difficulties of the SVMs consists of finding the correct parameters to train them. In our case, there are two parameters: the \(C\) from the maximum margin classifier and the \(\gamma\) from the Radial Basis Function kernel. The goal is to find the optimal values of the two parameters \(C\) and \(\gamma\) so that the classifier can accurately predict the unknown data. Very often, this is achieved through a cross-validation procedure by using a grid search for the
two parameters. In this work, the Parameter Estimation Algorithm [15] for
the binarization methods is employed for the detection of the correct SVMs
parameters. The stages of the algorithm for the detection of the best SVM
parameters values are:

Stage 1: The ranges of the SVM parameter values \([c_s, c_e]\) for the parameter
\(C\) and \([\gamma_s, \gamma_e]\) for the parameter \(\gamma\), are initialized: \(c_s = 0, c_e = 100, \gamma_s = 0\)
and \(\gamma_e = 10\).

Stage 2: The number of steps executed in each iteration for each parameter
is initialized: \(s_c = 10\) (\(C\) parameter) and \(s_\gamma = 10\) (\(\gamma\) parameter).

Stage 3: The lengths of each step are calculated according to the following
equations:

\[
L_c = \frac{c_e - c_s}{s_c - 1}\quad (for\ the\ parameter\ C) \quad (17)
\]
\[
L_\gamma = \frac{\gamma_e - \gamma_s}{s_\gamma - 1}\quad (for\ the\ parameter\ \gamma) \quad (18)
\]

Stage 4: All the values of parameters \(C\) and \(gamma\) are calculated for each
step according to the following equations:

\[
C (i) = s_c + k \cdot L_c, \forall k \in [0, s_c - 1] \quad (19)
\]
\[
\gamma (i) = s_\gamma + k \cdot L_\gamma, \forall k \in [0, s_\gamma - 1] \quad (20)
\]

Stage 5: The two pairs of the parameters values that give the best \((C_1, \gamma_1)\)
and second best result \((C_2, \gamma_2)\) are calculated by the cross - validation tech-
nique.

Stage 6: The ranges of the two parameters are redefined for the next it-
teration according to the following equations:

\[
[c_s', c_e'] = \begin{cases} [C_1, C_2] & \text{if } C_1 < C_2 \\ \left[\frac{c_s + C_1}{2}, \frac{c_e + C_2}{2}\right] & \text{if } C_1 = C_2 \\ [C_2, C_1] & \text{if } C_1 > C_2 \end{cases} \quad (21)
\]
\[
[\gamma_s', \gamma_e'] = \begin{cases} [\gamma_1, \gamma_2] & \text{if } \gamma_1 < \gamma_2 \\ \left[\frac{\gamma_s + \gamma_1}{2}, \frac{\gamma_e + \gamma_2}{2}\right] & \text{if } \gamma_1 = \gamma_2 \\ [\gamma_2, \gamma_1] & \text{if } \gamma_1 > \gamma_2 \end{cases} \quad (22)
\]
**Stage 7:** The steps of the new ranges for the next iteration are redefined according to the following equations:

\[
s'_{c} = \begin{cases} 
    s_{c} - 1 & \text{if } c_{c} - c_{s} \leq s_{c} \\
    s_{c} & \text{anything else}
\end{cases} \quad (23)
\]

\[
s'_{\gamma} = \begin{cases} 
    s_{\gamma} - 1 & \text{if } \gamma_{c} - \gamma_{s} \leq s_{\gamma} \\
    s_{\gamma} & \text{anything else}
\end{cases} \quad (24)
\]

**Stage 8:** if \(s'_{c} \cdot s'_{\gamma} \geq 5\) then the process is continued to Stage 3 and all the stages are repeated again with the new ranges and steps. If \(s'_{c} \cdot s'_{\gamma} < 5\) then the procedure is terminated and the best parameter values are those calculated at Stage 6 of the last iteration.

The values of the SVM parameters obtained by the above procedure were: \(C = 1.051\) and \(\gamma = 0.184\).

In order to evaluate the advantages of the Parameter Estimation Algorithm, the values of the SVMs parameters are detected also, with the cross-validation procedure by using a grid search. The results which are depicted at Table 2 shows that the Parameter Estimation Algorithm detects the parameter values speedier and more precise than the grid search.

<table>
<thead>
<tr>
<th>Procedure</th>
<th>(C) Parameter</th>
<th>(\gamma) Parameter</th>
<th>Cross-Validation Result</th>
<th>Calculation Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grid Search</td>
<td>5.345</td>
<td>0.345</td>
<td>83.234</td>
<td>4.73 hours</td>
</tr>
<tr>
<td>Parameter Estimation Algorithm</td>
<td>1.051</td>
<td>0.184</td>
<td>88.124</td>
<td>3.52 hours</td>
</tr>
</tbody>
</table>

**3.3. The Relevance Feedback algorithm**

Figure 5 illustrates the structure of the Relevance Feedback technique. When the system presents the initial retrieval results to the user, he is able to tag one or more images as wrongly or rightly retrieved. The system utilizes this information by grouping the descriptor of those word-images (including the original query descriptor) as training data for the SVMs. Then, all the words-images are rated between 0 and 100 based on the Eq. 16 and they are
presented to the user in descending order with respect to the corresponding rating.

The user is enabled to tag more word images based on the new retrieval results. The descriptors of the new tagged word images are added to the previous training data. Next, all the words-images are rerated based again on the Eq. 16 and the new retrieval results are presented to the user, again. This procedure is repeated until the retrieval results are satisfactory to the user.

Section 5.4 presents the impact of the aforementioned Relevance Feedback technique to the initial retrieval results for three different image databases.

4. Similarity Measure

For similarity matching, the distance $D_{(Q,S)}$ of two image descriptors $Q_{(i)}$ and $S_{(i)}$ is calculated using the proposed weighted Minkowski $L_1$ distance:

$$D_{(Q,S)} = 10 \cdot |Q_{(i)} - S_{(i)}| + \sum_{k=0}^{3} \sum_{n=1}^{10} ((11 - n) \times |Q_{(10 \times k + n + 1)} - S_{(10 \times k + n + 1)}|)$$

(25)

where $i$ is the number of the descriptor bin

The advantage of the above weight Minkowski $L_1$ distance is the utilization of the DCT ability to store more information in the first coefficients. For example, the similarity of the second DCT coefficient is more important than the similarity of the tenth coefficient. This hypothesis is proven correct by the experimental results (See Section 5.4). The above distance was found to be preferable than the normal Minkowski $L_1$, $L_2$ (Euclidian Distance), Bhattacharyya [16] and the non-binary Tanimoto coefficient [17] distances.
5. Evaluation

5.1. Introduction

In order to evaluate the general applicability of the CSPD, three distinct and different databases images are chosen. The first is the MPEG-7 CE1 Set B [6] database. It consists of 1400 shapes organized into 70 different groups taken from real word objects which have common nature shape distortions. It is created for the evaluation of a similarity-based retrieval system and its descriptors performance under a variety of shape distortions. The advantage of using a standard database is the easiness to compare the performance of the CSPD with other descriptors that are presented to the literature.


The third database consists of handwritten Greek documents that contains 173 words from various persons grouping into 29 different words using either lowerscases or upercases letters.

The next section describes the structure of the retrieval system used to test the proposed descriptor and Section 5.3 describes the experiments.

5.2. The Structure of the Retrieval System

The overall structure of the retrieval system that uses the CSPD is presented in Figure 6. It is constituted of two different parts: the Offline and the Online procedure. The system is implemented on the Microsoft .NET Framework 3.5. The programming language that is used is the C#/XAML. The web address of the implemented system is the http://orpheus.ee.duth.gr/cspd. Furthermore, the implementation of the proposed shape descriptor can be downloaded freely from the http://orpheus.ee.duth.gr/download/cspd.zip.

In the Offline operation, the images are analysed and the proposed descriptor is calculated for each one of them. Particular, for the handwritten documents a binarization method (Figure 7(b)) and a Connected Components Labelling and Filtering method [18] are employed to detect the initial word boundaries [19]. The binarization is achieved by applying the Sauvola technique [20] as it provides the best results [15].

In order to find the exact word limits the following steps are performed:

Step 1: The space between the characters is removed in order the expanded
Figure 6: The overall structure of the implemented Retrieval System. The red boxes (and italics font) are implemented for the handwritten documents only.
CCs of the characters to overlap (see Step 5 and 6).

**Step 2:** All the Connected Components (CCs) (Figure 7(c)) are identified.

**Step 3:** The maximum height of the CCs ($CC_{\text{max}}$) is calculated.

**Step 4:** In Kavallieratou et al. [19] has been proven that the height of a word can reach the fourth of a character mean size due to presence of ascenders and descenders. So, in order to remove noise and other unrelated objects all the CCs which their height or width is less than one quarter of the maximum height ($CC < \frac{CC_{\text{max}}}{4}$) are rejected (Figure 3(d)).

**Step 5:** The left and right sides of the CCs are expanded by $\frac{CC_{\text{max}}}{2}$ and the up and down sides by $\frac{CC_{\text{max}}}{4}$ as depicts the Figure 3(e). The CCs are expanded by half of the mean size left and right in order to merge non overlapping italics characters. As the space between the characters is removed it is certainly that if an overlapped CC exists then this is a character of the same word. The same logic applies to expansion of CCs to the up and down sides.

**Step 6:** Finally, the overlapping CCs are merged (Figure 7(f)).

Then, the CSPD for each shape/word is calculated and stored in a database. In the Online operation which is the only operation visible to the user, he/she selects a query image (shape or word) in order to find its similar images. The system uses the query descriptor that corresponds to the user selected image and it calculates the similarity measure with each image descriptor that residues in the database and presents the results. The user can tag some results as correct or wrong and the system is utilizing this information to correct the retrieve results.

5.3. **The Evaluation Metrics**

The objective *Averaged Normalized Modified Retrieval Rank* (ANMRR) [4] metric is employed to evaluate the performance of the word retrieval system that uses the proposed CSPD descriptor. The ANMRR is always in range of 0 to 1 and the smaller the value of this measure is, the better the matching quality of the query. ANMRR is the evaluation criterion used in all of the MPEG-7 colour core experiments. Evidence shows that the AN-
MRR measure approximately coincides linearly with the results of subjective evaluation of search engine retrieval accuracy [4]. To calculate the ANMRR metric, first the average rank $AVR(q)$ for query $q$ is calculated:

$$AVR(q) = \frac{NG(q)}{NG(q)} \sum_{k=1}^{NG(q)} \frac{Rank(k)}{NG(q)}$$  \hspace{1cm} (26)

where $NG(q)$ is the number of ground truth images for query $q$. In our case, the ground truth images are those that depict the same word. The $Rank(k)$ is the retrieval rank for the $k$ ground truth image:

$$Rank(k) = \begin{cases} R & \text{if } R \leq L \\ L + 1 & \text{if } R > L \end{cases}$$  \hspace{1cm} (27)

where $R$ is the position of the $k$th ground truth image for the query $q$.

$$L = \min \{X_{NG} \times NG(q), 2 \times \max \{NG\}\}$$  \hspace{1cm} (28)

and

$$X_{NG} = \begin{cases} 2 & \text{if } NG(q) > 50 \\ 4 & \text{if } NG(q) \leq 50 \end{cases}$$  \hspace{1cm} (29)
Next, the modified retrieval rank is calculated:

\[ MRR(q) = AVR(q) - 0.5 \times [1 + NG(q)] \]  

(30)

The \( MRR(q) \) is 0 in case of a perfect retrieval. The normalized modified retrieval rank is computed as follows:

\[ NMRR(q) = \frac{MRR(q)}{1.25 \times L - 0.5 \times [1 + NG(q)]} \]  

(31)

Finally the average of NMRR over all queries is defined as:

\[ ANMRR(q) = \frac{1}{Q} \sum_{q=1}^{Q} NMRR(q) \]  

(32)

where \( Q \) the total number of the queries/words. In our experimental databases the \( Q = 173 \) for the Greek Word Database, \( Q = 4860 \) for the George Washington database and \( Q = 1400 \) for the MPEG-7 CE1 Set B database.

In addition to the ANMRR metric, the Recall and the Precision metrics are used for the evaluation of the proposed retrieval system. Recall is the ratio of the number of relevant records retrieved to the total number of relevant records in the database. Precision is the ratio of the number of relevant records retrieved to the total number of irrelevant and relevant records retrieved. In our evaluation, the precision and recall values are expressed in percentage. The Average Precision (AP) for a single query is the mean of the all precision values for this single query and the Mean Average Precision is the mean of the all average precision values.

5.4. Experiments

**Evaluation of the coefficients number:** The first experiment analyzes the impact of the DCT coefficients number to the retrieval results. It was run at the MPEG-7 CE1 Set B database and all shapes in the set used as queries (total 1400 queries). Each feature is utilized alone as a descriptor. Figure 8 and Figure 9 depicts the MAP and the ANMRR respectively of the retrieval results starting from the first coefficient and increasing by one.

The evaluation measures demonstrate that the information that is stored after the eleventh DCT coefficient has not any impact to retrieval results so it is safe to disregard it.
Figure 8: The Mean Average Precision by the number of DCT coefficients that employed for (a) Vertical Projection (b) Horizontal Projection (c) Top Shape (d) Bottom Shape

Figure 9: The ANMMR by the number of DCT coefficients that employed for (a) Vertical Projection (b) Horizontal Projection (c) Top Shape (d) Bottom Shape
Evaluation of the quantization bits number: This experiment involves the effect of the quantized feature vectors bits number to the retrieval results. The Precision/Recall graph and the ANNMR are calculated for binary representation from one bit to 5 bits for the MPEG-7 CE1 Set B database. All shapes in the set used as queries (total 1400 queries). Figure 10 illustrate very clearly that 3bits/bin is enough to quantize the descriptor without impaired the retrieval results.

Similarity measure evaluation: Figure 11 depicts Precision/Recall and ANMRR scores for five similarity measures. These are: the proposed
Weighted Minkowski $L_1$, the Minkowski $L_1$, the Minkowski $L_2$, the Bhattacharyya and the non-binary Tanimoto coefficient. The experimental results show that the proposed similarity measure yields the best scores.

**Relevance feedback algorithm evaluation:** The fourth experiment involves the effect of relevance feedback algorithm to the retrieval results for the three databases. First the Precision/Recall graph and the ANNMR scores are calculated for the initial retrieval results. Next, the user tags the first two correct and the first two wrong word images and the Precision/Recall graph and NMRR values are recalculated. Finally a second iteration of the
Relevance Feedback Algorithm is evaluated. The user tags the next first two correct and wrong images and the evaluation metrics are calculated again. Figure 12 depicts the Precision/Recall and ANNM scores for each database.

**Evaluation against previous retrieval systems.** Towards the evaluation of the CSPD image document retrieval effectiveness, it is assessed against a previous Document Image Retrieval System through Word-Spotting (DIRStWS) [21] which it uses for database 100 noisy document images. For evaluations purposes 160 searches were made with random 160 query words. Figure 13 depicts the MAP and ANMRR for all the queries for each retrieval framework. Although, the DIRStWS uses lengthy word specific features (≈744 bytes), the proposed descriptor provides better retrieval results.

**Evaluation of the proposed descriptor against previous descriptors:** In order to evaluate the retrieval performance and the calculation time of the proposed descriptor, another seven various descriptors are employed. These are: the Geometric Moment Descriptor (GMD) [22], the Zernike Moment Descriptor (ZMD) [6, 23], the Grid Descriptor (GD) [24], the Curvature Scale Space Descriptor (CSSD) [23], the Fourier Descriptor (FD) [25], the Generic Fourier Descriptor (GFD) [26] and the Centroid Distance Fourier Descriptor (CDFD) [27]. These descriptors are implemented in the same framework environment as the proposed descriptors. That is the NET Framework 3.5 and the C# as the programming language.

Moreover, the Shape Context (SC) descriptor [28] is implemented in the same framework (C#/.NET 3.5) taking 200 samples for each shape using the Hungarian method for finding the best match. This descriptor is following a local-feature approach in addition to the above descriptors which they use global features. Local-feature approaches provide a slightly better retrieval effectiveness than global features [29]. They represent images with multiple points in a feature space in contrast to single-point global feature representations. While local approaches provide more robust information, they are more expensive computationally due to the high dimensionality of their feature spaces and usually need nearest neighbors approximation to perform points-matching [30].

The retrieval tests are conducted on the MPEG-7 CE1 Set B shape database which it consists of 1400 shapes of 70 groups. This standard database is very good for testing the overall robustness of the shape representation of the descriptors. Figure 14(a) depicts the Recall/Precision curves of the retrieval results and Figure 14(b) presents the Average Mean Precision of the curves.
Figure 12: (a) The Precision/Recall graph of the Initial retrieval results and for 2 RF iterations for the MPEG-7 CE1 Set B database. (b) The ANMRR scores of the Initial retrieval results and for 2 RF iterations for the MPEG-7 CE1 Set B database. (c) The Precision/Recall graph of the Initial retrieval results and for 2 RF iterations for the George Washington words database. (d) The ANMRR scores of the Initial retrieval results and for 2 RF iterations for the George Washington words database. (e) The Precision/Recall graph of the Initial retrieval results and for 2 RF iterations for the Greek Handwritten words database. (f) The ANMRR scores of the Initial retrieval results and for 2 RF iterations for the Greek Handwritten words database.
Furthermore, Table 3 shows the calculate time in order to extract the above descriptors from the shapes of the MPEG-7 CE1 Set B database. The experiment was tested on Windows platform of an Intel Core 2 6400 CPU utilizing both cores. As the above experiments show that the proposed descriptor (CSPD) manages to achieve better retrieval results from the other descriptors (except from the ZMD and the local feature descriptor SC) despite its small size and its smallest extraction time (Table 3). Only the ZMD and the SC perform better at the retrieval results but the calculation time and the size of these descriptor are considerably bigger than the proposed descriptor. Furthermore, the SC employs considerably more retrieval times as the nearest neighbours approximation is more computational demanding than a simple distance. Generally, the CSPD is suitable for retrieval systems which demand small size, good results and short calculation and retrieval times.

6. Conclusion

In this paper we proposed a descriptor that contains conventional contour and region shape features. Its main advantages are the very small size (only 123bits); its low computation cost and its general applicability without compromise its retrieval accuracy. We evaluated it against various descriptors using the MPEG-7 CE1 Set B database and the results were in par with other more heavy computational
Figure 14: (a) The Precision/Recall Graph for each descriptor. (b) The Average Mean Precision for each descriptor.
Table 3: Descriptors calculation time for 1400 shapes.

<table>
<thead>
<tr>
<th>Descriptor</th>
<th>Total Time of features extraction</th>
<th>Average time of feature extraction of each shape</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSPD</td>
<td>4396 (msec)</td>
<td>3.14 (msec)</td>
</tr>
<tr>
<td>GMD</td>
<td>40172 (msec)</td>
<td>28.69 (msec)</td>
</tr>
<tr>
<td>ZMD</td>
<td>232244 (msec)</td>
<td>165.889 (msec)</td>
</tr>
<tr>
<td>GD</td>
<td>141108 (msec)</td>
<td>100.792 (msec)</td>
</tr>
<tr>
<td>CSSD</td>
<td>16754 (msec)</td>
<td>11.958 (msec)</td>
</tr>
<tr>
<td>FD</td>
<td>11244 (msec)</td>
<td>8.027 (msec)</td>
</tr>
<tr>
<td>GFD</td>
<td>12195 (msec)</td>
<td>12.246 (msec)</td>
</tr>
<tr>
<td>GDFD</td>
<td>12034 (msec)</td>
<td>8.59 (msec)</td>
</tr>
<tr>
<td>SC</td>
<td>253678 (msec)</td>
<td>181.198 (msec)</td>
</tr>
</tbody>
</table>

or largest descriptors. Also, in order to test its general applicability three different databases are employed: A database that contains Greek handwritings words; a database that contains handwritten words from the documents of the George Washington and the MPEG-7 CE1 Set B database.

Also, a Relevance Feedback technique based on the SVMs while using the CSPD is presented. The parameters of the SVMs are calculated by a technique that it is inspired by the Parameter Estimation Algorithm for the detection of binarization methods parameters. The results appear to be remarkably improved, which shows both the CSPD ability of the increasing discrimination and the valid use of the SVMs decision function as similarity measure.

Finally, the advantage of the proposed descriptor is its balance between its size, its calculation time and its retrieval results. It is suitable for existing retrieval systems that they need to add shape information to their descriptors without compromising theirs storing and computational requirements.

7. Acknowledgments

This paper is part of the 03EΔ679 research project, implemented within the framework of the Reinforcement Programme of Human Research Manpower (PENED) and co-financed by National and Community Funds (25%) from the Greek Ministry of Development - General Secretariat of Research and Technology and (75%) from the E.U. - European Social Fund.
References


