COMBINING TEXTUAL AND VISUAL CLUSTERS FOR SEMANTIC IMAGE RETRIEVAL AND AUTO-ANNOTATION

Erbug Celebi*, Adil Alpkocak†
Dokuz Eylul University
Department of Computer Engineering
35100 Izmir, Turkey
*erbugelebi@gmail.com †alpkocak@cs.deu.edu.tr

Keywords
Image annotation, image retrieval, clustering, semantic, C3M

Abstract
In this paper, we propose a novel strategy at an abstract level by combining textual and visual clustering results to retrieve images using semantic keywords and auto-annotate images based on similarity with existing keywords. Our main hypothesis is that images that fall in to the same text-cluster can be described with common visual features of those images. In this approach, images are first clustered according to their text annotations using C3M clustering technique. The images are also segmented into regions and then clustered based on low-level visual features using k-means clustering algorithm on the image regions. The feature vector of the images is then changed to a dimension equal to the number of visual clusters where each entry of the new feature vector signifies the contribution of the image to that visual cluster. Then a matrix is created for each textual cluster, where each row in the matrix is the new feature vector for the image in that textual cluster. A feature vector is also created for the query image and it is then appended to the matrix for each textual cluster and images in the textual cluster that give the highest coupling coefficient are considered for retrieval and annotations of the images in that textual cluster are considered as candidate annotations for the query image. Experiments have demonstrated that good accuracy of proposal and its high potential of use in annotation of images and for improvement of content based image retrieval.

1. Introduction
The advances in multimedia technology and the rapidly expanding multimedia collections on the Internet have attracted significant research efforts in providing tools for effective retrieval and management of multimedia data. Image retrieval is based on the availability of a representation scheme of image content. Image content descriptors may be visual features such as color, texture, shape, and spatial relationships, or semantic primitives. Conventional information retrieval was based on text, and those approaches to textual information retrieval have been transformed into image retrieval in a variety of ways. However, “a picture is worth a thousand words”. Image contents are much more versatile compared with texts, and the amount of visual data is already enormous and still expanding very rapidly. Hoping to deal with these special characteristics of visual data, content-based image retrieval methods have been introduced. It has been widely recognized that the family of image retrieval techniques should become an integration of both low-level visual features addressing the more detailed perceptual aspects and high-level semantic features underlying the more general conceptual aspects of visual data. Neither of these two types of features is sufficient to retrieve or manage visual data in an effective or efficient way [5]. Although efforts have been devoted to combining these two aspects of visual data, the gap between them is still a huge barrier in front of researchers. Intuitive and heuristic approaches do not provide us with satisfactory performance. Therefore, there is an urgent need of finding the latent correlation between low-level features and high-level concepts and merging them from a different perspective. How to find this new perspective and bridge the gap between visual features and semantic features has been a major challenge in this research field.

There are so many researches in the literature on this subject: James Z. Wang et al. [4] present an image retrieval system, SIMPLicity, which uses integrated region matching based upon image segmentation. Their system classifies images into semantic categories such as textured/non-textured graph photograph. For the purpose of searching images, they have developed a series of statistical image classification methods. Duygulu et. al. and Hoffman have used a probabilistic approach to find latent classes of image corpus [3][6]. Sometimes, extracting semantics of images can be considered as auto-annotation of images with keywords. One approach to automatically annotate images is to look at the probability of associating words with image regions. Mori et. al.[11] used a co-occurrence model, which they look at the co-occurrence of words with image regions created using regular grid. More recently, a few other researches have also examined the problem using machine learning approaches. In particular Duygulu et. al.[3] proposed to describe images using a vocabulary of blobs. Many researchers have studied [6][11] about translating one language to an other by using probabilistic approaches called Probabilistic Latent Semantic Analysis (PLSA). Statistically-oriented approaches, believes that machines can learn (about) natural language from training data such as document collections and text corpora.

In this study our main hypothesis is that images that fall in to the same text-cluster can be described with common
visual features of those images. In this approach, images are first clustered according to their text annotations using C^3M clustering technique. The images are also segmented into regions and then clustered based on low-level visual features using k-means clustering algorithm on the image regions. The feature vector of the images is then changed to a dimension equal to the number of visual clusters where each entry of the new feature vector signifies the contribution of the image to that visual cluster. Then a matrix is created for each textual cluster, where each row in the matrix is the new feature vector for the image in that textual cluster. A feature vector is also created for the query image and it is then appended to the matrix for each textual cluster and images in the textual cluster that give the highest coupling coefficient are considered for retrieval and annotations of the images in that textual cluster are considered as candidate annotations for the query image.

The main contribution of this paper is to propose a new strategy (1) to retrieve images using semantic keywords and (2) auto-annotate images based on similarity with existing keywords for bridging the gap between low-level visual features and lack of semantic knowledge in multimedia information retrieval. Our solution works on an abstract level and combines both textual and visual clustering algorithms performance. The main idea behind this strategy is that the images within the same text cluster should also have same common visual features and could be stored in the same visual cluster. In our study, C^3M and k-mean clustering algorithms are used for clustering textual annotations of images and low level visual features, respectively.

The remainder of the paper is organized as follows: The next section gives a review of C^3M algorithm and usage of our strategy combining textual and visual clustering properties is discussed in detail in section 3. In section 4, experimentation results are presented and the last section 5 concludes the paper and provides an outlook to our future studies on this subject.

2. C^3M

Cover Coefficient-based Clustering Methodology (C^3M) is originally proposed by Can and Ozkarahan [2] to cluster text documents. The base concept of the algorithm, the cover coefficient (CC), provides a means of estimating the number of clusters within a document database and relates indexing and clustering analytically. The CC concept is used also to identify the cluster seeds and to form clusters with these seeds. The retrieval experiments show that the information-retrieval effectiveness of the algorithm is compatible with a very demanding complete linkage clustering method that is known to have good retrieval performance.

Cover Coefficient-based Clustering Methodology (C^3M) employs document clusters as cluster seeds and member documents. Cluster seeds are selected by employing the seed power concept and the documents with the highest seed power are selected as the seed documents. In their original work, Can and Ozkarahan, they showed that the complexity of C^3M is better than most other clustering algorithms, whose complexities range from $O(m^2)$ to $O(m^3)$. Also their experiments show that C^3M is time efficient and suitable for very large databases. Moreover, its low complexity is experimentally validated. C^3M has all the desirable properties of a good clustering algorithm.

C^3M algorithm is a partitioning type clustering (clusters cannot have common documents). A generally accepted strategy to generate a partition is to choose a set of documents as the seeds and to assign the ordinary (non-seed) documents to the clusters initiated by seed documents to form clusters. This is the strategy used by C^3M. Cover coefficient, CC, is the base concept of C^3M clustering. The CC concept serves to:

i. identify relationships among documents of a database by use of the CC matrix,
ii. determine the number of clusters that will result in a document database;
iii. select cluster seeds using a new concept, cluster seed power;
iv. form clusters with respect to C^3M, using concepts (i)-(iii);
v. Correlate the relationships between clustering and indexing.

C^3M is a seed-based partitioning type clustering scheme. Basically, it consists of two different steps that are cluster seed selection and the cluster construction.

$D$ matrix is the input for C^3M, which represents documents and their terms. It is assumed that each document contains $n$ terms and database consists of $m$ documents. The need is to construct $C$ matrix, in order to employ cluster seeds for C^3M. $C$, is a document-by-document matrix whose entries $c_{ij}$ ($1 < i, j < m$) indicate the probability of selecting any term of $d_i$ from $d_j$. In other words, the $C$ matrix indicates the relationship between documents based on a two-stage probability experiment. The experiment randomly selects terms from documents in two stages. The first stage randomly chooses a term $t_i$ of document $d_i$; then the second stage chooses the selected term $t_i$ from document $d_i$.

For the calculation of $C$ matrix, $c_{ij}$ one must first select an arbitrary term of $d_i$, say, $t_k$, and use this term to try to select document $d_j$ from this term, that is, to check if $d_j$ contains $t_k$. In other words, we have a two-stage experiment. Each row of the $C$ matrix summarizes the results of this two-stage experiment.

Let $s_{ik}$ indicate the event of selecting $t_k$ from $d_i$ at the first stage, and let $s'_{jk}$ indicate the event of selecting $d_j$, from $t_k$ at the second stage. In this experiment, the probability of the simple event “$s_{ik}$ and $s'_{jk}$” that is, $P(s_{ik} \cdot s'_{jk})$ can be represented as $P(s_{ik}) \times P(s'_{jk})$. To simplify the notation, we use $s_{ik}$ and $s'_{jk}$ respectively, for $P(s_{ik})$ and $P(s'_{jk})$, where:
\[
s_{ik} = \frac{d_{ik}}{\sum_{h=1}^{n} d_{ih}}, \quad \text{and} \quad s'_{jk} = \frac{d_{jk}}{\sum_{h=1}^{m} d_{hk}},
\]
where \(1 \leq i, j \leq m, 1 \leq k \leq n\)

By considering document \(d_i\), we can represent the \(D\) matrix with respect to the two-stage probability model. Each element of \(C\) matrix, \(c_{ij}\), (the probability of selecting a term of \(d_i\) from \(d_j\)) can be founded by summing the probabilities of individual path from \(d_i\) to \(d_j\),

\[
c_{ij} = \sum_{k=1}^{n} s_{ik} s'_{jk}
\]

This can be rewritten as;

\[
c_{ij} = \alpha_i \sum_{k=1}^{n} d_{ik} \beta_k d_{jk}, \text{ where } 1 \leq i, j \leq m
\]

Where \(\alpha_i\) and \(\beta_k\) are reciprocals of the \(i^{th}\) row sum and \(k^{th}\) column sum, respectively, as shown below;

\[
\alpha_i = \frac{1}{\sum_{j=1}^{m} d_{ij}}, \text{ where } 1 \leq i \leq m
\]

\[
\beta_k = \frac{1}{\sum_{j=1}^{m} d_{jk}}, \text{ where } 1 \leq k \leq n
\]

**Properties of \(C\) Matrix**

The following properties hold for the \(C\) matrix:

i. For \(i \neq j\), \(0 \leq c_{ij} \leq c_{ii}\) and \(c_{ii} > 0\)

ii. \(c_{i1} + c_{i2} + \cdots + c_{im} = 1\)

iii. If none of the terms of \(d_i\) is used by the other documents, then \(c_{ij} = 0\), otherwise, \(c_{ij} < 1\).

iv. If \(c_{ij} = 0\), then \(c_{ji} = 0\), and similarly, if \(c_{ij} > 0\), then \(c_{ji} > 0\), but in general, \(C\) is not symmetric.

v. \(c_{ij} = c_{ji}\) if \(d_i\) and \(d_j\) are identical.

From these properties of the \(C\) matrix and from the CC relationships between two document vectors, \(c_{ij}\) can be seen to have the following meaning:

\[
c_{ij} = \begin{cases} 
\text{extent to which } d_i \text{ is covered by } d_j \text{ for } i \neq j \\
\text{(coupling of } d_i \text{ with } d_j) \end{cases}
\]

\[
c_{ij} = \begin{cases} 
\text{extent to which } d_i \text{ is covered by itself for } i=j \\
\text{(decoupling of } d_i \text{ from the rest of the documents).} \end{cases}
\]

As can be seen from the foregoing discussions, in a \(D\) matrix, if \(d_i\) (\(1 \leq i \leq m\)) is relatively more distinct (i.e., if \(d_i\) contains fewer terms that are common with other documents), then \(c_{ij}\) will take higher values. Because of this, \(c_{ij}\) is called the decoupling coefficient, \(\delta_i\) of \(d_i\). (Notice that \(\delta_i\) is a “measure” of how much the document is not related to the other documents, and this is why the word coefficient is used.)

The sum of the off-diagonal entries of the \(i^{th}\) row indicates the extent of coupling of \(d_i\) with the other documents of the database and is referred to as the coupling coefficient, \(\psi_i\) of \(d_i\). From the properties of the \(C\) matrix,

\[
\delta_i = c_{ii} : \text{decoupling coefficient of } d_i
\]

\[
\psi_i = 1 - \delta_i : \text{coupling coefficient of } d_i
\]

\[
\delta = \frac{\sum_{i=1}^{m} \delta_i}{m}, \text{ where } 0 < \delta < 1
\]

\[
\psi = \frac{\sum_{i=1}^{m} \psi_i}{m}, \text{ where } 0 \leq \psi \leq 1
\]

### 3. Image Auto-Annotation

In our strategy to auto-annotate images, we assume that images with similar annotations must share, at least, some similar low-level features. If this is correct, can this correlation be used to associate some low-level visual features addressing the more detailed perceptual aspects and high-level semantic features? More formally, images that fall into the same textual cluster can be described with their common visual features. One can easily think that it is possible to find images with annotations as counter examples that do not obey the underlying hypothesis. However, it is not possible to say that images with similar annotation never shares similar low-level features. It is clear that our approach is highly depending on the training set, and images must be annotated with care. On the other hand, our main hypothesis relies on the intersected parts of both textual and low-level visual features.

In our approach, images are first clustered according to their text annotations. The images are also segmented into regions and then clustered based on low-level visual features on the image regions. The feature vector of the images is then changed to a dimension equal to the number of visual clusters where each entry of the new feature vector signifies the contribution of the image to that visual cluster. Then a matrix is created for each textual cluster, where each row in the matrix is the new feature vector for the image in that textual cluster. A feature vector is also created for the query image and it is then appended to the matrix for each textual cluster and images in the textual cluster that give the highest coupling coefficient are considered for retrieval and annotations of the images in that textual cluster are considered as candidate annotations for the query image.

#### 3.1 Training

Training phase of our approach based on the combination of textual and visual clustering and has three main steps: **textual clustering, visual clustering and replacing**. The first step occurs at the training phase of the system and all of training images, \(T\), are clustered according to their textual
annotated by using C'M. Secondly, all the image clusters are clustered according to visual similarities by k-means clustering algorithm where the number of cluster is \( n_{c-color} \) for color features. Let \( K(t) \) is the k-means function and \( T_i \) is set of regions, clustering can be formally defined as follows:

\[
K(t):T_i \rightarrow Mc_i \quad (0 \leq i \leq n_{c-color})
\]

where \( T=\{t: t \text{ is the segment of image } I, \forall I \in T \} \) holds the corresponding cluster id of region \( t \) for color clusters. The dimension of image feature vectors after \( K(t) \) transformation is equal to the number of elements in \( Mc_i \) (cluster sets). Then, each image, \( I_j \), is represented as a vector in \( n_{c-color} \) dimensional space.

\[
I_j = < i_{j1}, i_{j2}, \ldots, i_{jn_{c-color}} >
\]

Each entry of new feature vector signifies the contribution of corresponding color cluster to the image \( j \). Formally, let \( i_{jk} \) indicates the \( k^{th} \) entry of vector \( I_j \) which is for \( j^{th} \) image in collection. More formally, an arbitrary entry of vector \( I_j \) can be defined as follows:

\[
i_{jk} = \left\{ \begin{array}{ll}
\sum w_i & \text{if } K(s_i) = m_j, K(s_j) = m_k \text{ for } \forall s_i \in I_j, \forall s_j \in I_j, p \neq t \\
0 & \text{if } K(s_i) = m_j, K(s_j) = m_k \text{ for } \forall s_i \in I_j, \forall s_j \in I_j, p \neq t
\end{array} \right.
\]

The vector is normalized so that sum of the entries of vector \( I_j \) is equal to 1. In another words, in this step, each image is transformed into a dimension, called region space. We have constructed new feature vectors for each image in the training set by using k-means clustering. The new features for each image are consisting of cluster IDs that represent the segments of images. At the end of first two steps of training phase, we have two sets of clusters: first set is a set that contains the clusters of images based on text annotations and the second one contains clusters of images based on visual features of their regions.

The last step of training is replacing the image vectors of textual clusters with visual features. More clearly, we use textual clustering of images, but each image within the cluster is represented by visual features for annotation and retrieval. This concludes the training phase and forms a combination of textual and visual features of image collection. This is the most important phase of our approach, which is based on the hypothesis that images with similar annotations should also have similar low level features, and images that fall into the same text cluster should also have common visual features and could be stored in the same color cluster.

### 3.3. Combining Clusters

#### 3.3.1 Training Step

In this section we will give details about how we train the system and how we retrieve similar images and annotate the query image in our experiments. At the first stage of the training step, we have clustered the train images into clusters by considering their textual annotations. In this way conceptually similar images will be stored in the same cluster. C'M algorithm has been used to cluster images by their annotations because it is a partitioning type clustering (clusters cannot have common documents) and it doesn't create small and huge sized clusters. We have specified the text cluster size as 315 because \( n_{c-text} \) is 315. The maximum cluster size for C'M that doesn't generate zero-length cluster for our train set. At the second step, image regions (segments) are clustered in to 200 clusters (\( n_{c-color}=200 \)) according to their low-level features. Some researchers called each of these clusters as blobs [3].

Once the blobs are constructed a new feature \( m_{x,n_{c-color}} \) matrix is created where each entry, \( i_{jk} (1 \leq j \leq m, 1 \leq k \leq n_{c-color} \text{ color } m \text{ is } 4500) \) represents the weight of each color cluster (blob) in \( j^{th} \) image. For instance in the case of an image contains regions that fall in to \( 3^{rd}, 16^{th} \) and \( 37^{th} \) blob clusters, value of \( i_{3s}, i_{16s}, i_{37t} \) will be the percentage of those regions in the considered image respectively and rest of the vector elements will be equal to zero. In the case of different image region within the image fall in to the same color-cluster, their weights are summed.

#### 3.3.2 Query Step

Assume that we have been given a query image \( Q \) and asked to find out best annotations and/or retrieve similar images to \( Q, M \) values for the query image are evaluated according to \( K(t) \) function as in the training phase and the new feature vector \( q' \) is obtained for query image \( Q \). Once \( q' \) constructed it is assumed as a new member to each textual cluster by adding query vector to \( I \), clusters as the last document. In other words, the problem becomes to find out which of those clusters are most suitable to store the image \( q' \).

In order to find answer to this question, the correlation of the query image \( q' \) to each textual cluster is evaluated first. In another words, how query image is correlated with other images in each cluster is found out from C'M properties. As described in Section 2, C'M algorithm defines the correlation of \( q' \) with coupling coefficient (\( \psi_q \) of \( q' \) and the result of \( C \) matrices of each \( I' \). Coupling value gives how much an image is related to others within the same collection. In another words, an image that shares a lot of common features with the other documents has a high coupling, but low decoupling coefficient otherwise [12]. Diagonal entries of \( C \) Matrix show the decoupling
coefficient of each image with each other. We calculated the distance of each images to query image as follows:

\[
dist_i = \frac{1}{C_m} \cdot \frac{\sum_{j \neq i} \text{dist}_{ij}}{C_m - 1}
\]

for each image \(i\) in cluster \(C\) and where \(m\) is the number of images in cluster \(C\).

From these results, it is possible obtain ranked results and the 7 top image having the highest correlation, is chosen for annotation/retrieval. Then, the highest frequent 5 keywords selected from annotations of those images are selected as annotations of the query image.

4. Experiments

In this section we have described our experiments that are performed to assess the strengths and weakness of our system. We have used 4500 images from Corel image dataset to train the system and select 500 images that are distinct from training set to perform evaluations. In the image set, 10 largest regions are extracted from the images and each region is represented by 13 low-level features. We have obtained those feature sets from Duygulu et. al [3] which is publicly available. Usage of existing dataset allows us to compare the performance of similar models in the literature in a controlled manner.

4.1 Training and query

At the training phase, first, images are clustered according their text annotations with C'M. In our experiments C'M evaluates the \(n_{c-text}\) (number of clusters) as 89 for train set’s annotations. However, each image in train set is annotated with at least 1 and at most 5 keywords, C'M resulted with few huge clusters. Because of this issue, we have specified \(n_{c-text}\) as 315 that is the maximum number of clusters with non empty clusters. Secondly image regions are clustered according to their selected low level features with \(k\)-means. We selected number of clusters, \(n_{c-color}\) as 200 experimentally.

4.2 Image retrieval and auto-annotation

Whilst the query phase, images that are most similar to query image according to our proposed methodology are retrieved. Retrieved images are ranked and first 7 images are selected as query result. Annotations of retrieved images are selected as candidate annotations. We select 5, 7 or 10 (three distinct experiments) high frequent keywords from candidate annotations to auto-annotate the query image. A total of 260 one word queries are possible in the test dataset.

In our experiments we used precision and recall tests to evaluate the auto-annotation results. Precision and Recall tests for image retrieval is not an easy task in the absence of test beds for used image database. Also, it is not easy for auto-annotation, because of semantic similarity of keyword pairs such as \(\text{sun} \) and \(\text{sky}\), or \(\text{horse} \) and \(\text{mare}\). For that reason we need to find out the synonyms of keywords, if there is any, in the dataset to make the evaluations of results better.

We have constructed a thesaurus for the keywords used in the dataset with C'M. C'M matrix in C'M algorithm [2] is used to make term clusters. C'M matrix also gives information about term correlations in the considered data set, so it is meaningful to use the matrix to find similar terms (for the dataset) that will yield us synonym terms. For each keyword in the data set, we select a synonym keyword as follows with a threshold of 0.05:

\[
\text{Synonym}(\text{keyword}_i) = \text{keyword}_j \quad \text{where,} \quad \forall k, c'_{ik} = \max(c'_{i,j}).c'_{ik} < \text{threshold}
\]

Once we have constructed a thesaurus specific to the dataset, we modified the annotation of test images by adding the synonym of each keyword. Examples from the generated thesaurus can be seen at Table 1.

| ladder | buildings | vendor | people |
| crowd | people | shirt | people |
| African | people | polar | bear |
| nest | birds | arctic | fox |
| woman | people | Fawn | deer |
| Jet | plane | straightaway | cars |
| runway | plane | f-16 | jet |
| Boeing | plane | grizzly | bear |
| perch | birds | ocean | coral |
| hawk | branch | writing | sign |

Table 1: Examples of synonyms from Thesaurus generated with C'M.

4.3 Experimental Results

Similar to the previous studies on automatic image annotation, the quality of automatic image annotation is measured by the performance of retrieving auto-annotated images regarding to single-word queries. For each single word-query, precision and recall are computed using the retrieval results and original test image annotation in the dataset that is modified as described in the previous section. Single word queries are performed by first auto-annotating each image and performing single-word query on the annotation results. Accuracy of image auto-annotations will also mean accuracy on image retrieval because of; annotations are obtained from the retrieval results.

We have named our methodology as TSIS where it stands for “text space to images space” conversion. We have performed queries on all of images based on TSIS-5, TSIS-7 and TSIS-10 methodologies individually and shown in Table 2. As the results of our experiments we obtained that TSIS-10 (color features with 10 most frequent keywords) method has got better results among others when we use the thesaurus. But, we use TSIS-5 to compare our results with other studies since other researchers have used mostly 5 keywords to annotate the query image. Few of our auto-annotation results are as in Figure 3.
4.4 Model Comparison

We compare the annotation performance of the similar models in the literature where they have used the same data set as in our study. We annotate each test image with 5 keywords (TSIS-5) by using our methodology as in other similar studies. Figure 1 shows the results obtained on complete set of 260 words that appear in the test set. The values of recall and precision were averaged over the set of testing words, as suggested by [13, 14]. Figure 1 also shows the result, taken from [13][4], obtained with various other methods under the same experimental set. Specially we consider Co-occurrence Model [11], the Translation Model [3], Cross-Media Relevance Models (CMRM) [15], Multiple-Bernoulli Relevance Model (MBRM) [14] and Mix-Hier[13]. Figure 2 presents Precision-Recall graph for the proposed methodology.

MBRM and Mix-Hier have better performance than the method proposed, if we consider the recall values that are positive. On the other hand that is another important issue is the complexity of annotation process. In our experiments over the set of 500 test images, the average annotation time was 14 seconds where it is 268 seconds for Mix-Hier and 371 for MBRM.

Table 2: Experimentation results of TSIS-5, TSIS-7 and TSIS-10

<table>
<thead>
<tr>
<th># of words with Recall&gt;0</th>
<th>TSIS-5</th>
<th>TSIS-7</th>
<th>TSIS-10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Co-occ.</td>
<td>76</td>
<td>83</td>
<td>94</td>
</tr>
<tr>
<td>Trans.</td>
<td>19</td>
<td>49</td>
<td>66</td>
</tr>
<tr>
<td>CMRM</td>
<td>122</td>
<td>137</td>
<td></td>
</tr>
<tr>
<td>MBRM</td>
<td>137</td>
<td>137</td>
<td></td>
</tr>
<tr>
<td>MixHier</td>
<td>137</td>
<td>137</td>
<td></td>
</tr>
<tr>
<td>TSIS</td>
<td>137</td>
<td>137</td>
<td></td>
</tr>
</tbody>
</table>

Figure 1. Performance comparison on the task of automatic image annotation on the Corel dataset.

Figure 3. Auto-annotation of images 113067, 152059, 153056, 22013, 122098 and 142057 with TSIS-7
5. Conclusion and Future Works

In this study, we proposed a new approach to semantically retrieve images using keywords and auto-annotate images based on similarity with existing annotated images. Our main hypothesis is that images that fall in the same text cluster, can be described with common visual features of those images. The system is highly relies on the overlapping of the similar parts of an image in both textually and visually although this hypothesis seems to strong and work on only for constrained image set. We have show that our proposal is capable to be used in auto-annotation of images and improve the retrieval effectiveness.

The system was trained with a test-bed containing 4500 images from COREL image database and tested with 500 images from outside the training database. Experiments have demonstrated that good accuracy of proposal and its high potential use in auto-annotation of images and for improvement of content-based image retrieval.

In our experiments we use 1 to 5 keywords to annotate images. This is not the usual case, where the content of an image is described by free text. In such case, high frequent keywords should be discarded from the documents by using stemming algorithms (i.e. Porter Stemming Algorithm) at first. Keywords that are not eliminated by stemming algorithm can be used to specify documents. Once keywords assigned to documents our methodology can be used without any change by starting from clustering documents with C'M.

In this study we have used only color features as the low level descriptors with constant parameters. We have been working on the performance improvement of our solutions under different parameters. We plan to work different number of clusters and observe the results in our future works. In addition to different parameters as well as considering the conditional probabilities of keyword occurrence. In longer term, we expect this solution to lead us into new researches including semantic web, semantic indexing, and development of image ontology automatically and extend to video.

References