Searching, Browsing, and Annotating Image Databases

Seminario per il Corso di Sistemi Informativi L-S

Ilaria Bartolini
MultiMedia DataBase Group
DEIS
i.bartolini@unibo.it

Bologna, May 12th, 2010
“Our scientific activity is focused on finding, within very large data bases, those objects and data that are better suited to fulfill the information needs of non-expert users. This leads us to consider both effectiveness (i.e. "quality" of results) and efficiency aspects of the search process, in order to scale it to large data volumes. Moreover, we also take into account the need to provide the user with simple but powerful tools, able to smooth the processes of query creation/customization and of result interpretation.

Until now, we applied our techniques to a plethora of different media types (images, text documents, time series, etc.), to genetic data and to patterns obtained through Data Mining processes (associative rules, clusters, etc.)”

Bologna, May 12th, 2010
MultiMedia DataBase Group

http://www-db.deis.unibo.it/MMDBGroup/

- Prof. Paolo Ciaccia
- Prof. Marco Patella
- Dr. Ilaria Bartolini

- Similarity-based image retrieval (Windsurf, PIBE, Imagination e Scenique)
- User preferences
- Indexing multimedia data (M-tree)
- Complex queries
- Approximate queries

Bologna, May 12, 2010
Outline

- Content-based image retrieval and browsing
- The semantic gap problem
- Content&Semantic-based image retrieval and browsing
- Semi-automatic techniques for image annotation
First approach to search for images relies on standard text-based techniques, provided objects come with a precise textual description of what they represent/describe, i.e., of their semantics.

However, the “annotation” of images is a subjective, time consuming, and tedious process (completely manual!!)

A more convenient approach, suitable to manage large DBs, is to automatically extract from images a set of (low-level) relevant numerical features that, at least partially, convey some of the semantics of the objects.

Clearly, which are the “best” features to extract depend on the specific application at hand (i.e., what we are looking for).

Look for cheetahs? This is fine; but, how to find it?
Content-based image similarity search

- Once we have feature values, we can search images by using them
- Assume we have a DB with N images and, for each of the N objects, we have extracted the “relevant features”
  - E.g., we could extract some color information

- We can now search for images whose feature values are “similar” (in some sense to be defined) to the feature values of our query [SWS+00, LSD+06, LZL+07, DJL+08]
The region-based approach

- **DB population time:**
  - Preprocess images to segment them into regions
  - Represent regions as vectors of features

- **Query time:**
  - Compare query regions to DB regions
  - Assess similarity between images by combining similarity between regions
Windsurf [ABP99, BCP00, BP00, BC03, Bar09a, BCP+09, BCP10]

- **Windsurf: Wavelet-Based Indexing of Images Using Regions Fragmentation**
  - **Discrete Wavelet Transform (DWT):** extracts a set of features representing the image in the color-texture space
  - **Clustering:** fragments the image into a set of regions using wavelet coefficients
  - **Similarity Features:** used to compare regions

I. Bartolini - Bologna, May 12th, 2010
Discrete Wavelet Transform (DWT)

- **Haar wavelet**: simple and quick
- Each coefficient is defined by:
  - level DWT ($l$)
  - frequency sub-band ($B$)
  - color channels ($H, S, V$)

\[
 w_j^{l;B} = \{ w_0^{l;B}, w_1^{l;B}, w_2^{l;B} \}
\]

\[ B \in \{L, LH, HL, HH\} \]
DWT: Practical example

I. Bartolini - Bologna, May 12th, 2010
Clustering (1)

- **K-means algorithm (3rd level and low frequency info)**
  - Choose \( k \) initial centroids;
  - Associate each point to its nearest centroid;
  - Recompute centroids and repeat previous step;
  - Stop when solution does not change.

- **Mahalanobis distance**:

  \[
  \delta \left( w_i^{3,LL}, w_j^{3,LL} \right) = \left( w_i^{3,LL} - w_j^{3,LL} \right)^T \Sigma^{-1} \left( w_i^{3,LL} - w_j^{3,LL} \right) 
  \]

  - Correlation between wavelet coefficients takes into account variations in color, i.e. **texture**

- **No spatial information**
Clustering (2)

- Optimal value for $k$?
- Minimization of a *validity* function
  - Intra-cluster distance
  - Clusters’ size
  - Inter-cluster distance

Input image

Clusters for $k=2$  Clusters for $k=10$  Clusters for $k=4$ (Optimal solution)

I. Bartolini - Bologna, May 12th, 2010
Similarity features

- Region similarity with Bhattacharyya distance
  - Regions are ellipsoids in 37-D feature space (all frequencies info is used)
    - (3-D centroid + 6-D covariance matrix + 1-D region size)
  - Distance between regions’ centroids (color info)
  - Covariance matrices (texture info)

\[
d_B(R_i, R_j) = \frac{1}{2} \ln \left( \frac{C_{R_i}^{3;B} + C_{R_j}^{3;B}}{\frac{1}{2} \cdot C_{R_i}^{3;B}} \cdot \frac{1}{2} \cdot C_{R_j}^{3;B} \right) + \frac{1}{8} \left( \mu_{R_i}^B - \mu_{R_j}^B \right) \cdot \left( \frac{C_{R_i}^{3;B} + C_{R_j}^{3;B}}{2} \right)^{-1} \cdot \left( \mu_{R_i}^B - \mu_{R_j}^B \right)
\]

I. Bartolini - Bologna, May 12th, 2010
Image similarity

- Similarity between images is a function of similarities among “matched” regions.
- How regions are "matched" can therefore strongly influence the result of a query:
  - “one-to-one” match (formulated as Assignment Problem)
  - “many-to-many” match (formulated as Transportation Problem)
Assignment problem

- Goal: “Find the optimal match where unit elements of fixed size are matched individually”

- Implemented with the Hungarian algorithm, maximizing a function that is monotonic in the similarity scores (e.g. average)

```
<table>
<thead>
<tr>
<th></th>
<th>r1</th>
<th>r2</th>
<th>r3</th>
<th>r4</th>
<th>r5</th>
</tr>
</thead>
<tbody>
<tr>
<td>q1</td>
<td>52</td>
<td>.17</td>
<td>.41</td>
<td>.16</td>
<td>.29</td>
</tr>
<tr>
<td>q2</td>
<td>.27</td>
<td>.19</td>
<td>.81</td>
<td>.35</td>
<td>.49</td>
</tr>
<tr>
<td>q3</td>
<td>1.0</td>
<td>.11</td>
<td>.27</td>
<td>.24</td>
<td>.29</td>
</tr>
</tbody>
</table>
```

```
<table>
<thead>
<tr>
<th></th>
<th>r1</th>
<th>r2</th>
<th>r3</th>
<th>r4</th>
<th>r5</th>
</tr>
</thead>
<tbody>
<tr>
<td>q1</td>
<td>.52</td>
<td>.17</td>
<td>.41</td>
<td>.16</td>
<td>.29</td>
</tr>
<tr>
<td>q2</td>
<td>.27</td>
<td>.19</td>
<td>.81</td>
<td>.35</td>
<td>.49</td>
</tr>
<tr>
<td>q3</td>
<td>1.0</td>
<td>.11</td>
<td>.27</td>
<td>.24</td>
<td>.29</td>
</tr>
</tbody>
</table>
```

\[
\frac{(.52+.81+1.0)}{3}=.77
\]

\[
\frac{(.29+.81+1.0)}{3}=.7
\]
Transportation problem

- Goal: “Find the least expensive flow where variable-size pieces of “mass” are allowed to be moved together”

- Implemented with the Earth Mover’s Distance (EMD):
  - “Given two distributions (let’s see one as piles of earth and the second as a collection of holes), EMD measures the least amount of work needed to fill the holes with earth. A unit of work corresponds to transporting a unit of earth by a unit of ground distance”
Index-based query processing

- Goal: to speed-up query evaluation time over sequential scan
  - reduce the number of images on which the region matching problem has to be solved
- DBAM (M-tree [CPZ97])
  - Good for k-nearest-neighbor (K-NN) query
  - Able to perform a sorted access to the data
- M-tree (i.e., “metric” tree) requires the (dis-)similarity function \( d(I_1, I_2) \) to be a metric:
  - \( d(I_1, I_2) > 0 \)
  - \( d(I_1, I_1) = 0 \)
  - \( d(I_1, I_2) + d(I_2, I_3) \geq d(I_1, I_3) \)
Query processing algorithms

- Indexing regions
  - Bhattacharyya distance is a metric
    - 1-1 matching (Hungarian)
    - M-N matching (EMD)
  - Monotonic scoring functions (e.g., avg) / qualitative preferences (e.g., Skyline) [BCC+06, BCO+07, BCP10]

- Indexing images
  - “If the ground distance (i.e., Bhattacharyya) is a metric and the total weights of the two distribution are equal, EMD is a metric” [BCP10]

- Full and partial queries [BCP10]
1-Nearest Neighbor query example

- 1-1 match
- Avg as scoring function

\[
\begin{align*}
\theta &= 0.91 \\
\theta &= 0.81 \\
\theta &= 0.725 \\
\end{align*}
\]

\[
\begin{align*}
{s_R} &= 0.92 & {s_I} &= 0.56 \\
{s_R} &= 0.74 & {s_I} &= 0.67 \\
{s_R} &= 0.73 & {s_I} &= 0.71 \\
{s_R} &= 0.82 & {s_I} &= 0.76 \\
{s_R} &= 0.72 & {s_I} &= 0.68 \\
\end{align*}
\]

I. Bartolini - Bologna, May 12th, 2010
“St. Peter” query

I. Bartolini - Bologna, May 12th, 2010
Visual results: flat visualization

I. Bartolini - Bologna, May 12th, 2010
Visual results: spatial visualization
Effectiveness comparison example

“Flowers” query  Windsurf clusters  Blobworld [CTB+99] clusters
WARP [BCP05]

- **WARP**: Accurate Retrieval based on Phase
  - Innovative Fourier-based approach for matching and retrieving similar shape
    - Exploitation of the phase of the Fourier coefficients
      - More accurate boundary description than using only amplitude
    - Use of the Dynamic Time Warping (DTW) distance
      - Good match even in presence of phase shifting
  
  ![Diagram](image.png)

I. Bartolini - Bologna, May 12th, 2010
FeedbackBypass [BCW00, BCW01]

- New approach to interactive similarity query processing
  - Increases the performances of traditional relevance feedback techniques [RHO+98]; it complements the role of relevance feedback engines by storing and maintaining the query parameters determined during the feedback loop over time

- We realized two implementations of FeedbackBypass:
  - The first one is based on Wavelet
  - The second one uses Support Vector Machine (SVM)
Content-based image browsing

- Till now we have implicitly assumed that the user “knows”
  - what s/he is looking for
  - how to formulate her queries/preferences
    - E.g., Query By Example (QBE) paradigm
- In some cases the user does not know at all what to look for; in these cases a “browsing” activity should be supported
  - to determine a good starting point for searching
  - to get an overall view of the DB contents
  - to give the user the ability to organize a collection of images (e.g. a personal photo album) in a semi-automatic way
Flat browsing example
Spatial browsing example
PIBE [BCP06, BCP07]

- **PIBE**: Personalizable Image Browsing Engine
- A novel **adaptive** image browsing engine
  - customizable hierarchical structure called *Browsing Tree* (BT)
  - graphical **personalization actions** to modify the BT
  - “local” reorganization of the DB
    - specific similarity criteria for each portion (sub-tree) of the BT
  - user customizations persist across different sessions
Principles of PIBE

- Three main ingredients behind the BT:
  - image **descriptors** (e.g. color histograms)
    - as points in a N-dimensional space
  - **(dis)similarity** functions (e.g. weighted Euclidean)
    - instance of a parameterized class of functions to support personalization actions
  - clustering algorithms (e.g. **k-means**)
    - BT is a hierarchical structure derived from a **hierarchical** clustering algorithm or, alternatively, by **recursively** applying a **partitioning** algorithm

- PIBE is **parametric** with respect to above choices
Browsing Tree (BT)

- PIBE uses:
  - 32-D HSV color histograms ($p$)
  - weighted Euclidean distances

$$d(p, q; w) = \left( \sum_{i=1}^{32} w_i (p_i - q_i)^2 \right)^{1/2}$$

- \textit{k-means} algorithm applied to the whole image DB and, recursively, to each of the derived $k$ clusters producing a initial BT ($w_i = 1$)

- each node of the BT corresponds to a cluster $C_j$ of images and maintains the
  - centroid $c(C_j)$ of $C_j$
  - representative image $p(C_j)$ of $C_j$ defined as

$$p(C_j) = \arg \min_p d(p, c(C_j); w_j); p \in C_j$$

- local weight vector $w_j$ computed as

$$w_{j,i} \propto 1/\sigma_{j,i}^2$$

I. Bartolini - Bologna, May 12\textsuperscript{th}, 2010
Browsing modalities

- Two modalities:
  - **vertical**: the user selects a representative image on the display and **zooms in** the cluster content
  - **horizontal**: the user explores regions of the space where **no representative** image is present
Visual results: vertical browsing
Visual results: horizontal browsing
Personalization actions

- $C_s$ is merged with $C_t$; $C_t$ is then reclustered

- $C_j$ is partitioned by using selected images; resulting subtrees are item clustered and used to replace $C_j$

- $C_j$ is partitioned by using selected images; resulting subtrees are clustered and used to replace children of $C_j$
Visual example of BT personalization

Initial BT

Custom BT
Outline

- Content-based image retrieval and browsing
- The semantic gap problem
- Content&Semanitc-based image retrieval and browsing
- Semi-automatic techniques for image annotation
The semantic gap problem

- Characterizing the image content by means of low level features (e.g., color, texture, and shape) represents a completely automatic solution to image retrieval
  - However low level feature are not always able to properly characterize the semantic content of images
    - E.g., two images should be considered “similar” even if their semantic content is completely different

- This is due to the semantic gap existing between the user subjective notion of similarity and the one according to which a low level features-based retrieval system evaluate two images to be similar
  - prevent to reach 100% precision results
Possible solution

- (Semi-)automatically provide a **semantic characterization** (e.g., by means of keywords/tags) for each image able to capture its content
  - E.g., ([sky, cheetah] vs. [sky, eagle])

- **Combine** visual features with tags by taking the best of the two approaches [LSD+06, LZL+07, DJL+08]
Outline

- Content-based image retrieval and browsing
- The semantic gap problem
- Content&Semantic-based image retrieval and browsing
- Semi-automatic techniques for image annotation
Scenique [BC08b, Bar09]

- Scenique: Semantic and Content-based Image Querying
  - Image retrieval and browsing system that profitably exploits both low level features and manually and/or (semi-)automatically associated textual annotations
  - Based on a simplified version of the multi-structural framework [FKK+05] which allows objects, (i.e., images in our case) to be organized into a set of orthogonal dimensions, also called facets

“I. Bartolini, SEBD 2009

“Photos of animals I took during my summer vacations”
Principles of Scenique

- Provides the user with two basic facilities:
  1) an image annotator (…we will see in few minutes!!), that is able to predict new tags for images, and
  2) an integrated query facility which allows the user to search and browse images exploiting both visual features and tags
     - possibly organized in visual and semantic facets
     - take the form of trees
     - default semantic facet to ensure compatibility with systems/devices that do not consider any tag organization (e.g., Flickr)
Semantic facets

- Node labels are tags, with the root node be tagged with the facet name.
- The same tag can appear in different facets as well as in different nodes of the same facet.
- Each tag in a tree corresponds to a path in the tree (i.e., semantic tag).
Visual facets

- Built upon **low-level features** (e.g., color, texture, and shape) à la PIBE
  - that the user can refine

Each node in a tree corresponds to a **cluster** of similar visual feature and is labeled using a representative photo

A photo is not forced to be part of a visual facet
Technical details

- **Feature-based retrieval**
  - Images as a set of regions represented by means of color and texture features (à la *Windsurf*)

- **Semantic-based retrieval**
  - Multiple tags associated to photos
  - *WordNet* as lexical ontology [*Miller 1995*]
    - ISA relation
  - Semantic “relaxation”
    - Semantic similarity criterion to compare terms [*Lin 1997*]

- **Integration policies**
Semantic relaxation

Problem:
“*If I am looking for bear photos but neither bear nor specialized term of bear images are present in the DB (or the cardinality is too low)… I will get empty/unsatisfactory result!!*” 😞

- Semantic relaxation as a weight associated to each level of the hierarchy (starting from the query tag)
  - percentage of relaxation the user is willing to accept

The similarity between two tags (e.g., “brown bear” and “feline”) is computed as:
\[
Sim(t_1, t_2) = \frac{2 \times \text{level(common-father)}}{\text{level}(t_1) + \text{level}(t_2)}
\]

- The similarity between a tag and tags in its sub-tree is equal to 1
  - e.g., \( sim(bear, brown bear) = 1 \)
Faceted-oriented query

tag: “Flower”
...with semantic relaxation

tag: “Flower”

Vascular Plant

Seed Plant

Flowering Plant

Flower

Petunia

Orchid

Poppy

Bush

Rose

Semantic Similarity (“Flower”, “Rose”) =

\[
\frac{2 \times \text{level(“Vascular Plant”)}}{\text{level(“Flower”)} + \text{level(“Rose”)}}
\]

I. Bartolini - Bologna, May 12th, 2010
Faceted-oriented + content query

I. Bartolini - Bologna, May 12th, 2010
Integration policies

1) Semantic similarity

<table>
<thead>
<tr>
<th>Query image</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
</table>

(Chroma similarity = null)

2) Content similarity

<table>
<thead>
<tr>
<th>Query image</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
</table>

(Semantic similarity = null)
Query visual example

“I want images of bear from the Arctic Ocean that look like the provided one”
Query visual example
Browsing visual example
Browsing visual example
Browsing visual example
Browsing visual example
Browsing visual example
Browsing visual example
Browsing visual example

I. Bartolini - Bologna, May 12th, 2010
Browsing visual example
Browsing visual example
Outline

- Content-based image retrieval and browsing
- The semantic gap problem
- Content&Semantic-based image retrieval and browsing
- Semi-automatic techniques for image annotation
Image annotation

- Given an image, which are the **labels** that better describe its content?

  ![Image 1](horse, grass, ground) ![Image 2](sky, bridge, river, House of Parliament, Big Ben)

- **Manual** process
  - tedious
  - time consuming
  - user subjectivity
- Need to define (semi-)**automatic** techniques
Automatically infer semantic to images

The automation of the annotation process requires user intervention

1) Relevant feedback
   - Exploiting user feedback to understand which are real relevant images to the query: e.g., SCENIQUE + FeedbackBypass

2) Learning
   - The system is trained by means of a set of images that are manually annotated by the user (training phase)
   - Exploiting the training set, the system is able to predict labels for uncaptioned images: the test image is compared to training images; labels associated to the “best” images are proposed for labeling (labeling & testing phases)
Imagination [BC08a]

- **Imagination**: IMAGE (semi-)automatic annotation
- Images as set of regions (à la Windsurf)

- Labels are *tags* à la *Flickr*, which are associated at the image level
- *Graph-based* approach (à la *Page Rank*)
  - 3-level of graph objects
    - Images
    - Regions with low level features (i.e., color and texture)
    - Tags assigned to images
  - plus *K*-NN links computed on region similarities

"*Given a new image provide tags that are affine to the image and semantically correlated to each other*"
Intuitive example

regions

DB images

tags

deer, grass, bush

bear, rock, grass, water, ground

rock, bear, grass, water, ground

new image

?, ..., ?
Mixed Media Graph (MMG) construction

$G_{MMG}$

DB images:
- deer, grass, bush
- bear, rock, grass, water, ground
- rock, bear, grass, water ground

New image:
Random Walk with Restart [PYF+04]

- restart at the query node (with probability $p$)
- randomly walk to one link (with probability $1-p$)
- For each tag node a relative frequency is computed approximating the steady state probability
Why tags correlation?

- **MMG + RWR** heavily relies on NN edges involving the new image (i.e., low level features)
  - If a region of the new image is highly similar to a region of $G_{MMG}$, which however has some terms unrelated, this might easily lead to having such tags highly scored! 😞
    - *Uncorrelated* tags, or even *contradictory*
- **MMG + RWR** returns a fixed number ($PT$) of tags 😞
Analyzing correlations of tags

- Link analysis on a sub-graph of $G_{MMG}$ to find highly-correlated tags
  - bipartite graph $G_T$
  - second-order bipartite graph $G_T^2$
“an edge between nodes \((I_i, I_j)\) and \((T_r, T_s)\) is added iff the two edges \((I_i, T_r)\) and \((I_j, T_s)\), equivalently, \((I_i, T_s)\) and \((I_j, T_r)\), are in \(G_T\)”
PSimRank algorithm [FR05]

- A similarity score is computed for each tags node of $G^2_T$
  - “two nodes are similar if they are referenced by similar nodes”
    - two tags are similar if they are present in similar images
    - two images are similar if they contain similar tags
- The process is independent from the query node
  - off-line

<table>
<thead>
<tr>
<th>Tags</th>
<th>Similarity [0,1]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tags</td>
<td></td>
</tr>
</tbody>
</table>

$I_1, I_2$ nodes

Edges $E(u)$

Nodes $u$
Putting it all together

- PT tags with highest *steady state probability* returned by MMG + RWR step are reduced considering tags correlation
- We model the problem as an instance of the *Maximum Weight Clique Problem* [BBP+99]
Imagination user interface
Predicted tags

![Image of an interface showing predicted tags with categories like FISH, MAMMAL, DESCAN, BLOSSOMS, and FOOD. The image also shows a sheep and grass, with options to assign terms and use a wordnet.](Image)
Annotation visual example within Scenique

“I want to annotate from scratch the selected image”
N.B. Agli studenti particolarmente interessati e appassionati agli argomenti illustrati, ricordo che su tali tematiche abbiamo diversi progetti di ricerca attivi e che c’è quindi largo spazio per svolgere sia attività di tipo preparatorio alla tesi che tesi di laurea specialistica.