2D Images Map Warping for Improved User Interaction

Daniele Borghesani, Costantino Grana, Rita Cucchiara
Dipartimento di Ingegneria dell'Informazione
University of Modena and Reggio Emilia, Italy
name.surname@unimore.it

Abstract

In this paper, we suggest an interaction model designed to fit users' expectations in front of an image retrieval system. A lightweight relevance feedback strategy, working directly on the 2D projection of image features, allows the user to spatially navigate the media collection maintaining the real-time constraint. A preliminary evaluation of this relevance feedback strategy shows good performance compared with other known approaches.

1. Introduction

At the present time no multimedia retrieval system is so successful to represent a breakthrough into the habits of users in front of a computer. The best results in bringing multimedia retrieval to the masses have been achieved by Google, with their "find similar" or "filter by color" features in the image section of their search engine. The problem is the significant gap between the research view of such systems and the user perspective, which is strongly influenced by the way information is presented. Current interfaces do not allow the exploration of large image collections in a user centric manner, rather we are assisting to a standardization of interface solutions towards a grid-based layout where user is limited to look and scroll for more images. We believe that this approach is flawed, because it does not convey visual feedback about the content of the collection (nullifying all the multidimensional similarity relations between images) and it does not dynamically react to user's feedbacks.

In this paper, we suggest a user interface design capable of visualizing the similarity relations and the effect of relevance feedbacks from the original feature space into a two-dimensional mapping. This procedure allows the system to show to the user a real-time feedback of his manipulations, bringing him into the collection itself. In order to convey a physical meaning of visual objects, we also employed an attraction/repulsion modeling based on the lower or higher visual similarity between images.

2. Related work

The literature about data visualization is countless. The classical spatial arrangement of images is their placement on a grid, typically in row-major ordering based on relevance. Despite its simplicity, this visualization is unable to convey information on the structure of the collection, for example the availability of a cluster of similar images. An inspiring shift towards a new direction of UIs has been provided by Santini *et al.* [5]. In their paper, authors proposed a formalization and some new ideas on image semantics, highlighting how the placing of images and the direct manipulation through physical metaphors can help the user in the process of interacting with the image database, improving the retrieval capabilities of the system. A good review of visualization techniques has been proposed by Plant *et al.* in [4], which talked about mapping-based techniques (PCA, MDS, FastMap, SOM, ISOMAP, SNE, LLE), clustering-based techniques and graph-based techniques (mass-spring, pathfinder networks, NN^k networks). The idea of exploiting interactivity to support retrieval was extensively confirmed in the comprehensive survey of Heesch [3] which underlined the fundamental notion that the semantic gap can be filled with the interactivity and a user interface devoted to browsing by design. Our approach wants to follow these directives, suggesting the use of the map itself as a source of information to compute relevance feedback and propose visualization feedback to users.

3. System Architecture

In Fig.1 an overview of the system architecture is depicted. Notice that most of the common functional

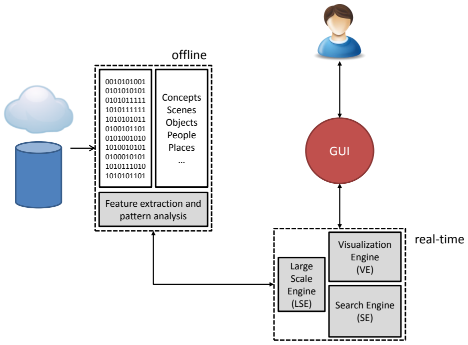

Figure 1. Overview of the relationships between multimedia systems and users.

blocks available in every multimedia retrieval system are here assumed as black boxes, to be customized as desired. At the center of the schema, we depicted what is usually considered a second-order problem, the user interface. In fact, following the flow of information which starts from users' intention down to the algorithms to satisfy the request, the first player is the interface itself. The interface is connected to the underlying processing, which can be basically represented with a Visualization Engine (VE), a Search Engine (SE) and a Large Scale Engine (LSE). VE uses the information processed by SE to correctly emphasize the relevant information. If very large databases are in use, LSE employs approximated techniques to support SE in nearest neighbor searches. Intuitively, SE is linked to two distance matrices. The first one is related to the visualization and it is manipulated directly by the user with relevance feedbacks. The second one instead regards the original feature space: it is generally computed offline, and it is exploited by the VE. The multimedia content is gathered locally (the private collections of users) or remotely (online repositories). Features are extracted from this content, and used offline to provide a first layer of high level semantic subdivision of content.

4. Implementation

The first problem to solve is the initial mapping of images on a two-dimensional space, as the computer screen is. Many techniques can be exploited, and good results can be quite easily obtained with a lot of them, but we have severe real time and interactivity constraints to consider for this application. Therefore, we need an algorithm capable of rapidly show a mapping, potentially in an incremental way, with the possibility to modify it supporting insertion or deletion of visual objects. Among the alternatives, we found out a promising solution in the FastMap [2]. To accomplish a two-dimensional mapping, the algorithm initially requires the selection of two couples of pivot objects (one for each axis of the projection). For this selection, authors proposed to heuristically choose a random element in the dataset, then selecting the farthest one o_a as first pivot, and conversely choosing o_b as the farthest from o_a as second pivot. In this paper, instead, we proposed an evolutionary solution based on Cuckoo Search [7] which allows a smarter selection of best pivots without impacting significantly on performance. We start from a dataset of N feature vectors $V = v_1, \ldots, v_N$, in which each v_i has dimensionality n based on the employed feature. A distance function $\mathcal{D}$ is defined between any two elements v_i and v_j. We want to obtain a two-dimensional mapping $O = o_1, \ldots, o_N$ for each feature vector in V. Initially, we need to find $M = 4$ distant-enough objects v_a, v_b, v_c and v_d (a *solution*), which will likely cover the most significant information trends within the dataset. Consider that we allow defining pivot objects which are not in the dataset itself.

Algorithm 1 Cuckoo Search inspired choose-distant-objects

```
Given an objective function $f(x) = \sum_{i}^{M} \sum_{j}^{M} \mathcal{D}(v_i, v_j)$ to maximize
Generate $T$ possible solutions (nests)
Set the maximum amount $G$ of generations $g$ (loops)
while $(g < G)$ or $(stopcriterion)$ do
    Randomly select a solution $k$ and add Gaussian noise
    Evaluate its fitness $F_k$
    Select a nest $h$ randomly
    if then$(F_h < F_k)$
        Replace the solution in $h$ with the new solution $k$
    end if
    Sort all possible solutions based on fitness $f$
    Remove a fraction of the worse nests
    Pass the best one to the next generation
end while
```

The pivot selection is detailed in Algorithm 1. This approach creates a set of T solutions for G successive generations. For each generation g_i, each solution is composed by a set of $M = 4$ potential pivots, for which an objective function (defined as the sum of relative distances) has to be maximized. The best solution is continuously passed to the next generation g_{i+1}, until the maximum number of generations G or another stop criterion is reached.

Once the pivots have been defined, given any triple v_a, v_i, v_b, we can extract the x_i coordinate of the mapping o_i of v_i by using Cosine Law as in Eq.4:

$$x_i = \frac{\mathcal{D}\left(v_a, v_i\right)^2 + \mathcal{D}\left(v_a, v_b\right)^2 - \mathcal{D}\left(v_b, v_i\right)^2}{2\mathcal{D}\left(v_a, v_b\right)} \quad (1)$$

Notice that the computation of x_i only needs the distances between objects, which are given or can be fastened using approximate techniques. Then the y_i coordinate of o_i is computed by means of the Pythagorean theorem. Firstly we define the distance $\mathcal{D}'$ on the hyperplane perpendicular to the line $\overline{o_a, o_b}$ exploiting the original distance $\mathcal{D}$ by means of Eq.2:

$$\mathcal{D}^{'}\left(v_i, v_j\right)^2 = \mathcal{D}\left(v_i, v_j\right)^2 - \left(x_i - x_j\right)^2 \quad (2)$$

Then, $\mathcal{D}^{'}$ is used as new distance function exploiting the Cosine Law as in Eq. to compute the y_i coordinate.

At this point, we detail how the user can interact with the obtained two-dimensional mapping in order to submit visual queries and relevance feedbacks. In particular, in our design we defined two search modalities. A *single query mode* in which the user wants to satisfy a precise search intention against the entire dataset, and a *multiple queries mode* in which the user wants to shape the visualization over multiple search intentions (i.e. for annotation purposes). Let $Q = q_k, \ldots, q_K$ the set of K queries potentially required by the user, and let F_{q_k} the set of feedbacks provided by the user for query q_k at incremental refinement steps, with $F^+_{q_k} \subset F_{q_k}$ the positives and $F^-_{q_k} \subset F_{q_k}$ the negatives. The goal of the algorithm is to visually minimize the effort required by the user to collect all the pictures with the same visual content.

4.1 Single query mode

1. *Query submission*
The user visually selects a query $q_k \in O$. The algorithm warps the two-dimensional space according to this unique search intention, which has no feedbacks F_{q_k} associated to it. The unit vector $\hat{o}_i$ connecting each image o_i to the current query q_k is extracted, then the distance in the original n-dimensional feature space is used in a ranking-based manner, scaling the image position along $\hat{o}_i$ proportionally to the ranking itself. In this way, the algorithm will focus the results around this query, getting similar results closer and dissimilar ones falling away from it.

2. *Query refinement*
The user can refine the query by selecting, hopefully within the neighborhood, those images matching or not his personal search intention (respectively good and bad feedbacks). At this point, the algorithm processes the new position of objects on the map scaling the projected objects along the unit vector, similarly to the previous step. Given d^+ the distance to the closest positive feedback in $F^+_{q_k}$ and d^- to the farthest negative in $F^-_{q_k}$, the new scaling factor d_i for each object o_i is computed as follows:

$$d_i = \mathcal{D}\left(o_i, q_k\right)\left(1 + \frac{d^+ - d^-}{\max(d^+, d^-)}\right) \quad (3)$$

The equation states that what is similar to positive prototypes should be moved towards the query, while what is similar to negative prototypes should be pushed away.

4.2 Multiple queries mode

1. *Queries submission*
The user proposes to the system his multiple search intentions (which kind of visual content he wants to highlight in the visualization) by selecting from the two-dimensional mapping a list of interesting queries Q. He also has the possibility to displace them on the screen as he likes, maximizing the inter-query distances.

2. *Queries refinement*
The user can drag positive samples in the neighborhood of each query and, as before, he can select which one is not influential for the given cluster. These relevance feedbacks increase the influence of each cluster with respect to the images on the map sharing the same visual content, and conversely increase the repulsion over the most dissimilar ones. The strength in attraction or repulsion of each cluster has been modeled in a way directly proportional to the amount of feedbacks, following a Coulomb's law like formulation. The normalized distance function $\mathcal{D}$ is transformed by means of a sigmoid function in order to determine if an attraction occurs (the picture on the map is sufficiently similar to a particular cluster) or conversely a repulsion occurs.

3. *Influence update*
Each cluster wields an influence upon the other ones. The influence is modeled as a sum of force vectors whose magnitude is directly proportional to the affinity to each cluster. In this way, clusters tend to be segregated to each other, generating different partitions on the map, easily allowing the

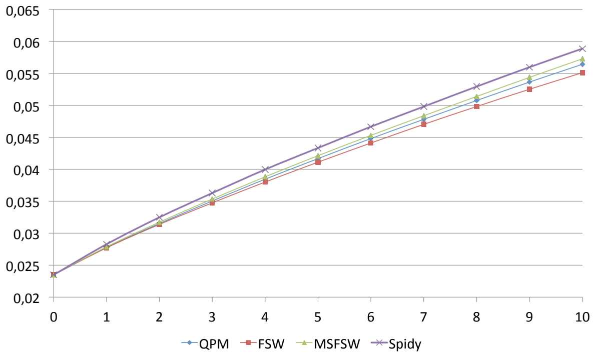

Figure 2. Single query mode relevance feedback results on training partitions of Sun397.

user to highlight the differences of the visual content he was looking for.

5. Experimental results

Regarding the single query scenario, we conducted our preliminary tests on Sun397 scene dataset [6]: we exploited the 10 training partitions proposed by authors, averaging the results over roughly 198500 queries. As visual descriptor, we employed a simple 512-bin RGB histogram: we would like to stress that we are not going to analyze the retrieval performance in absolute terms, but only to highlight the relevance feedback performance over time in comparison with other typical approaches in the same context. An automated procedure has been defined reporting the mAP with 10 subsequent iterations for each image taken as a query. The results are reported comparing our approach with standard *Query Point Movement* (QPM) and *Feature Space Warping* (FSW) approaches, as well as the *Mean Shift Feature Space Warping* (MSFSW) [1] as an hybrid of the previous techniques. As shown in Fig.2, our relevance feedback strategy, despite working solely on the two-dimensional projection, evidences good performance, comparable to the other proposals working on the full n-dimensional feature space. The approximation we introduced allows to satisfy the real-time constraints as well as the need for the user to interact directly with the visual content and see the outcome of his visual searches. Notice that both FSW and MSFSW would require a full space transformation at each interaction, becoming infeasible at increasing dataset sizes. In the multiple queries scenario (Fig.3), we had the opportunity to conduct only some qualitative evaluations, collecting opinions of the system working on a variety of different datasets (images downloaded from Google Images, artistic datasets, person datasets for forensics). As a preliminary analysis, we can conclude that an alternative view to the usual 2D grid, offering users the perception of relative similarities and direct manipulation capabilities, have been nicely welcomed. A more precise user test is under assessment.

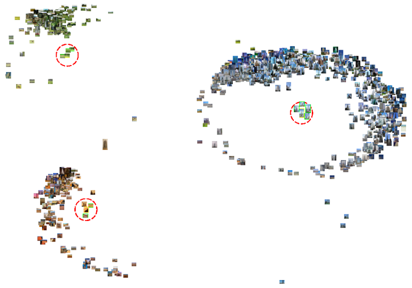

Figure 3. A screenshot of the application.

6. Conclusions

This paper suggests a new approach to incorporate a lightweight relevance feedback strategy directly into a smart visualization procedure in order to improve the browsing proficiency of the user through interaction. The proposed technique could have a good impact on the ability of the user to deal with large amounts of multimedia information.

References

[1] Y. Chang, K. Kamataki, and T. Chen. Mean shift feature space warping for relevance feedback. In *IEEE International Conference on Image Processing*, pages 1849–1852, 2009.

[2] C. Faloutsos and K. Lin. FastMap: a fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In *ACM SIGMOD International Conference on Management of Data*, pages 163–174. ACM, 1995.

[3] D. Heesch. A survey of browsing models for content based image retrieval. *Multimedia Tools and Applications*, 40:261–284, 2008.

[4] W. Plant and G. Schaefer. Visualising image databases. In *MMSP*, pages 1–6. IEEE, 2009.

[5] S. Santini, A. Gupta, and R. Jain. Emergent semantics through interaction in image databases. *IEEE Trans. Knowl. Data Eng.*, 13(3):337–351, 2001.

[6] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 3485–3492, 2010.

[7] X.-S. Yang and S. Deb. Cuckoo search via lévy flights. In *World Congress on Nature Biologically Inspired Computing*, pages 210–214, Dec. 2009.