Neurocomputing 166 (2015) 26–37

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

# The effect of low-level image features on pseudo relevance feedback

Wei-Chao Lin [a], Zong-Yao Chen [b], Shih-Wen Ke [c], Chih-Fong Tsai [b,*], Wei-Yang Lin [d]

[a] Department of Computer Science and Information Engineering, Hwa Hsia University of Technology, New Taipei, Taiwan
[b] Department of Information Management, National Central University, Taoyuan, Taiwan
[c] Department of Information and Computer Engineering, Chung Yuan Christian University, Taoyuan, Taiwan
[d] Department of Computer Science and Information Engineering, National Chung Cheng University, Chia-yi, Taiwan

ARTICLE INFO

*Article history:*
Received 21 October 2014
Received in revised form
2 April 2015
Accepted 6 April 2015
communicated by: Ke Lu
Available online 23 April 2015

*Keywords:*
Image retrieval
Relevance feedback
Pseudo relevance feedback
Feature representation
Rocchio algorithm

ABSTRACT

Relevance feedback (RF) is a technique popularly used to improve the effectiveness of traditional content-based image retrieval systems. However, users must provide relevant and/or irrelevant images as feedback for their queries, which is a tedious task. To alleviate this problem, pseudo relevance feedback (PRF) can be utilized. It not only automates the manual component of RF, but can also provide reasonably good retrieval performance. Specifically, it is assumed that a fraction of the top-ranked images in the initial search results are pseudo-positive. The Rocchio algorithm is a classic approach for the implementation of RF/PRF, which is based on the query vector modification discipline. The aim is to reproduce a new query vector by taking the weighted sum of the original query and the mean vectors of the relevant and irrelevant sets. Image feature representation is the key factor affecting the PRF performance. This study is the first to examine the retrieval performances of 63 different image feature descriptors ranging from 64 to 10426 dimensionalities in the context of PRF. Experimental results are obtained based on the NUS-WIDE dataset which contains 22156 Flickr images associated with 69 concepts. It is shown that the combination of color moments, edges, wavelet textures, and locality-constrained linear coding of the bag-of-words model provides the optimal feature representation, giving relatively good retrieval effectiveness and reasonably good retrieval efficiency for Rocchio based PRF.


## 1. Introduction

Advances in computer and multimedia technologies have allowed the production of digital images and the creation of large repositories for image storage with little cost. This has led a rapid increase in the size of image collections for multi-fold purposes, including digital libraries, medical imaging, art and museum collections, journalism, advertising, home photo archives, and so on. Clearly, it is now necessary to design automated image retrieval systems which can operate on a large scale.

The traditional image retrieval approach is based on manual image indexing with keywords assigned to images by human indexers during the database creation stage. Relevant images can be retrieved by using the indexed keywords as queries. However, there are some limitations to manual indexing. For example, it is a very time-consuming and expensive process, especially when the size of the image collection is very large, e.g., hundreds of thousands of images [24]. In addition, different indexers may assign different keywords to the same images, or the same indexers may perform differently given different circumstances and different times. In addition, during retrieval, users may not be aware of or agree with the indexed keywords or terms for queries which can lead to unsatisfactory retrieval results.

Content-Based Image Retrieval (CBIR) [30], which was proposed in the early 1990s, is a technique for automatically indexing images by extracting (low-level) visual features, such as color, texture, and shape. The retrieval of images is based solely upon the indexed image features. Therefore, it is hypothesized that *relevant* images can be retrieved by calculating the similarity between the low-level image contents through browsing, navigation, query-by-example, and so on. Typically, images are represented as points in a high dimensional feature space. Then, a metric is used to measure the degree of dis/similarity between images in this space. Thus, images corresponding closely to the query are classified as *similar* to the query and retrieved. Although CBIR introduced automated image feature extraction and indexation, it did not overcome the so-called *semantic gap* which is described in greater detail below.

The semantic gap is the gap between the computer extracted and indexed low-level features and the high-level concepts (or semantics) of a user's queries. In other words, the automated CBIR systems do not allow ready matching to the users' requests. The notation of similarity in the user's mind is typically based on high-level abstractions, such as

* Corresponding author at: Department of Information Management, National Central University, Taoyuan, Taiwan. Tel.: +886 3 422 7151; fax: +886 3 4254604.
*E-mail address:* cftsai@mgt.ncu.edu.tw (C.-F. Tsai).

http://dx.doi.org/10.1016/j.neucom.2015.04.037
0925-2312/

activities, entities/objects, events, or some evoked emotions, among others. In this situation, retrieval by similarity using low-level features like color or shape will not be very effective. In other words, human similarity judgments do not obey the requirements of the similarity metric used in CBIR systems. In addition, general users usually find it difficult to search or query images by using color, texture, and/or shape features only. They tend to prefer textual or keyword-based queries since these are easier to use and allow their information needs to be represented more intuitively [30].

One method applied to solve the semantic gap problem that affects the retrieval effectiveness of CBIR systems is relevance feedback (RF) [1,23]. RF is a technique used in traditional text-based information retrieval systems, namely a (supervised active learning) process intended to improve the system performance by refining the results of the original queries. The main idea, which is the so-called user-in-the-loop version, is to ask users to provide positive (relevant) and negative (irrelevant) examples as feedback on the initially retrieved document sets. RF can also be applied in image retrieval so that some retrieved images can be selected to receive positive or negative feedback for query refinement. In principle, RF is based on learning a set of 'optimal' feature weights for a query, or moving the query point toward the relevant images [40].

However, the duration of the iterative process required to meet the users' needs varies depending on the relevance feedback algorithms

**Table 1**
The 69 concepts.

| Airport (s) | Dancing | House (s) | Road (s) | Tower (o) |
|---|---|---|---|---|
| Animal | Dog (o) | Lake (s) | Rocks (o) | Town (s) |
| Beach (s) | Earthquake | Leaf (o) | Running | Train (o) |
| Bear (o) | Elk (o) | Military | Sand (o) | Tree (o) |
| Birds (o) | Fish (o) | Moon (s) | Sign (o) | Valley (s) |
| Boats (o) | Flags (o) | Mountain (s) | Sky (s) | Vehicle (o) |
| Bridge (s) | Flowers (o) | Nighttime (s) | Snow (s) | Whales (o) |
| Buildings (s) | Food | Ocean (s) | Soccer | Window (s) |
| Cars (o) | Fox (o) | Person | Sports | Zebra (o) |
| Castle (s) | Frost (s) | Plane (o) | Statue (o) | |
| Cat (o) | Garden (s) | Plants (s) | Street (s) | |
| Cityscape (s) | Glacier (s) | Police | Sun (o) | |
| Clouds (s) | Grass (s) | Railroad (s) | Swimmers | |
| Coral (o) | Harbor (s) | Rainbow (s) | Temple (s) | |
| Cow (o) | Horses (o) | Reflection (s) | Tiger (o) | |

**Table 2**
The 63 different feature representations.

| Low-level features | Abbreviation (dimensions) |
|---|---|
| **Color** | CH[a](144-D); CC[b](73-D); CM[c](225-D) |
| **Edge** | EDH[d](73-D) |
| **Texture** | WT[e](128-D) |
| **Bag-of-words** | BoW (500-D); LLC[f] (500-D); FV[g](10000-D) |
| **color+edge** | CH+EDH (137-D); CC+EDH (217-D); CM+EDH (298-D) |
| **color+texture** | CH+WT (192-D); CC+WT (272-D); CM+WT (353-D) |
| **color+BoW** | CH+BoW (564-D); CH+LLC (564-D); CH+FV (10064-D); CC+BoW (644-D); CC+LLC (644-D); CC+FV (10144-D) CM+BoW (725-D); CM+LLC (725-D); CM+FV (10225-D) |
| **edge+texture** | EDH+WT (201-D) |
| **edge+BoW** | EDH+BoW (573-D); EDH+LLC (573-D); EDH+FV (10073-D) |
| **texture+BoW** | WT+BoW (628-D); WT+LLC (628-D); WT+FV (10128-D) |
| **color+edge+texture** | CH+EDH+WT (265-D); CC+EDH+WT (345-D); CM+EDH+WT (426-D) |
| **color+edge+BoW** | CH+EDH+BoW (637-D); CH+EDH+LLC (637-D); CH+EDH+FV (10137-D); CC+EDH+BoW (717-D); CC+EDH+LLC (717-D); CC+EDH+FV (10217-D); CM+EDH+BoW (798-D); CM+EDH+LLC (798-D); CM+EDH+FV (10298-D) |
| **color+texture+BoW** | CH+WT+BoW (692-D); CH+WT+LLC (692-D); CH+WT+FV (10192-D); CC+WT+BoW (772-D); CC+WT+LLC (772-D); CC+WT+FV (10272-D); CM+WT+BoW (853-D); CM+WT+LLC (853-D); CM+WT+FV (10353-D) |
| **edge+texture+BoW** | EDH+WT+BoW (701-D); EDH+WT+LLC (701-D); EDH+WT+FV (10201-D) |
| **color+edge+texture+BoW** | CH+EDH+WT+BoW (765-D); CH+EDH+WT+LLC (765-D); CH+EDH+WT+FV (10265-D); CC+EDH+WT+BoW (845-D); CC+EDH+WT+LLC (845-D); CC+EDH+WT+FV (10345-D); CM+EDH+WT+BoW (926-D); CM+EDH+WT+LLC (926-D); CM+EDH+WT+FV (10426-D) |

[a] CH: Color histogram.
[b] CC: Color correlogram.
[c] CM: Color moments.
[d] EDH: Edge direction histogram.
[e] WT: Wavelet texture.
[f] LLC: Locality-constrained linear coding.
[g] FV: Fisher vector.

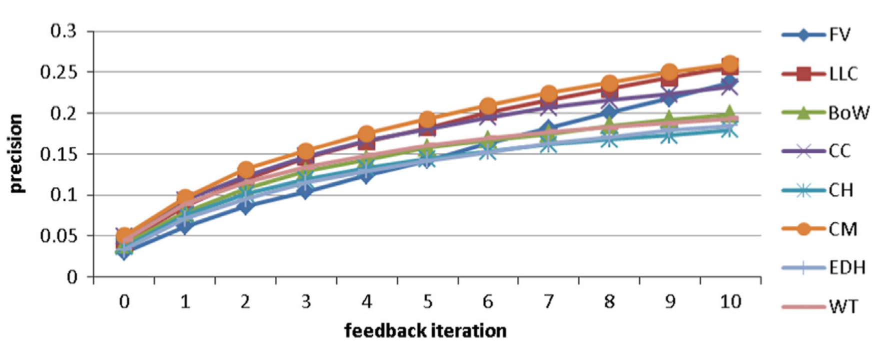


**Fig. 1.** Performance with single features.