ImageHunter: a novel tool for Relevance Feedback in Content Based Image Retrieval

Roberto Tronci, Gabriele Murgia, Maurizio Pili, Luca Piras, and Giorgio Giacinto

Abstract Nowadays, a very large number of digital image archives is easily produced thanks to the wide diffusion of personal digital cameras and mobile devices with embedded cameras. Thus, each personal computer, personal storage unit, as well as photo-sharing and social-network web sites, are rapidly becoming the repository for thousands, or even billions of images (i.e., more than 100 million photos are uploaded every day on the social site Facebook[1]). As a consequence, there is an increasing need for tools enabling the semantic search, classification, and retrieval of images. The use of meta-data associated to images solves the problems only partially, as the process of assigning reliable meta data to images is not trivial, is slow, and closely related to whom performed the task. One solution for effective image search and retrieval is to combine content-based analysis with feedbacks from the users. In this paper we present Image Hunter, a tool that implements a Content Based Image Retrieval (CBIR) engine with a Relevance Feedback mechanism. Thanks to a user friendly interface the tool is especially suited to unskilled users. In addition, the modular structure permits the use of the same core both in web-based and stand alone applications.

1 Introduction

The growing number of digital data such as text, video, audio, pictures or photos is pushing the need for tools allowing the quick and accurate retrieval of information from data. Whereas the results of traditional text data search methods are quite sat-

Roberto Tronci · Gabriele Murgia · Maurizio Pili
AMILAB - Laboratorio Intelligenza d'Ambiente, Sardegna Ricerche, loc. Piscinamanna, I-09010 Pula(CA), Italy, e-mail: {roberto.tronci, gabriele.murgia, maurizio.pili}@sardegnaricerche.it

Luca Piras · Giorgio Giacinto
DIEE - Department of Electric and Electronic Engineering, University of Cagliari, Piazza d'Armi snc, I-09123 Cagliari, Italy, e-mail: {luca.piras, giacinto}@diee.unica.it

[1] http://blog.facebook.com/blog.php?post=403838582130

isfactory, the same can not be said for visual or multimedia data. So far, the most common method for image retrieval is predicated on adding meta-data to the images as keywords, tag, label or short descriptions, so that the retrieval can occur through such annotations. The manual cataloguing of images, even though it requires expensive work and a large amount of time, is often not so effective. Describing a picture in words is not always easy, and the relevance of the description is strictly subjective.

By now, all mobile phones are equipped with cameras, and thanks to the Internet, social networks and almost "unlimited" storage space, the exchange of photos and digital images has become frenetic, to say the least. As a consequence there is an increasing need for tools enabling the semantic search, classification, and retrieval of images. As above-mentioned, the use of meta-data associated to the images solves the problems only partly, as the process of assigning meta data to images is not trivial, slow, and closely related to the persons who performed the task. This is especially true for retrieval tasks in very highly populated archives, where images exhibit high variability in semantic. It turns out that the description of image content tends to be intrinsically subjective and partial, and the search for images based on keywords may fit users' needs only partially. For this reason, since the early nineties, the scientific community focused on the study of Content Based Image Retrieval [9, 15, 11, 5] that it is based on the idea of indexing image by using low-level features such as color, texture, shape, etc.. Another difficulty in devising effective image retrieval and classification tools is given by the vast amount of information conveyed by images, and the related subjectivity of the criteria to be used to assess the image content. In order to capture such subjectivity, image retrieval tools may employ the so called *relevance feedback* [13, 18]. Relevance feedback techniques involve the user in the process of refining the search. In a CBIR task in which the RF is applied, the user submits to the system a query image, that is an example of the pictures of interest; starting from the query, the system assigns a score to the images in the database, the score being related to a similarity measure between the images and the query. A number of best scored images are returned to the user that judges them as relevant or not. This new information is exploited by the system to improve the search and provide a more accurate result in the next iteration. Faced with this new scenario, it has become increasingly urgent to find a way to manage this heap of data, to permit an effective search and to involve the user in this task.

Image Hunter is a full content-based image retrieval tool which does not need a text query in contrast to the vast majority of other applications [14, 2]. It is able to retrieve an ensemble of "similar" images from an image archive starting from an image provided by a user. Image Hunter is further equipped with a learning mechanism based on the relevance feedback paradigm that allows dynamically adapting and refining the search. In addition, the adaptability of the system has been enforced by the concurrent use of twelve different feature sets including color based, texture, and shape global descriptors.

The rest of the paper is organized as follows. Section 2.1 illustrates the core of the application, and its connections between the different modules. Section 2.2 briefly reviews the integrated learning process and relevance feedback mechanisms implemented in the application. Section 2.3 shows the graphical interface and explains how it works. Experimental results are reported in Section 3. Conclusions are drawn in Section 4.

2 Image Hunter

With the aim of building a practical application to show the potentialities of Content Based Image Retrieval tools with Relevance Feedback, we developed Image Hunter. This tool is entirely written in JAVA, so that the tool is machine independent. For its development, we partially took inspiration from the LIRE library [12] (that is just a feature extraction library). In addition, we chose Apache Lucene[2] for building the index of the extracted data.

Image Hunter is made up of two main parts: the core, and the user interface.

2.1 Image Hunter's core

The main core of Image Hunter is a full independent module in order to allow the development of a personalized user interface. The core is subdivided into four parts:

- Indexing
- LIRE interface
- Lucene interface for data storing
- Collection Search and Relevance Feedback

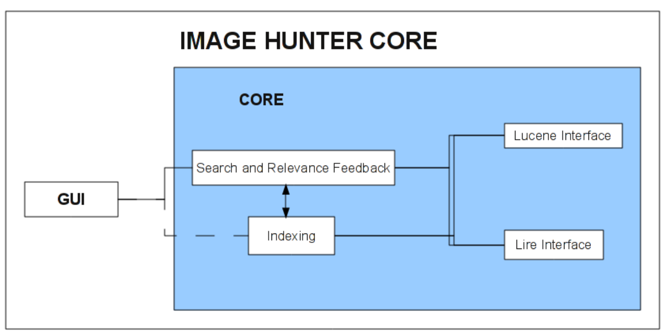

Fig. 1 Image Hunter's Core

The "Indexing" part has the role of extracting the visual features from the images. The visual features and other descriptors of the images are stored in a particular structure defined inside Image Hunter. For each image collection, all the data are stored in a database built according the Apache Lucene standard. Lucene turned out

[2] http://lucene.apache.org/

to be well suited for the storage needs of Image Hunter, as it resulted faster than other SQL based solutions. In particular, the indexes created by the use of Lucene can be easily moved by copying the folder that contains the index. In this way it is also quite simple to build a "portable" version of Image Hunter. Moreover, we had also adapted some of the main classes defined by Lucene to better fit to our needs: e.g., we created some methods to simplify the index administration, and we enriched the functionality of document manipulations.

Finally, we describe the "Collection Search and Relevance Feedback" module that is the more important as it implements the core engine of our system. Each time a user submits an image to be used as a *visual query*, the system computes the visual similarity between the query and each image in the collection. This visual similarity is computed in terms of the average of the normalized distances in each feature space. Then, the user can label the images provided by the system as relevant to her search or not, and the system exploits this feedback to learn which is the best combination of visual features that represents the semantic meaning that the user is associating to the query. Thus, in the feedback elaboration process, the visual similarity is computed in terms of a weighted combination of the distances in different feature spaces, rather than in terms of the average distance. In the following sections we describe the Relevance Feedback techniques implemented in Image Hunter (Section 2.2), and the web-based user interface that we have developed (Section 2.3).

2.2 Relevance Feedback techniques implemented in Image Hunter

In this section the three relevance feedback techniques implemented in the core are described. Two of them are based on the nearest-neighbor paradigm, while one of them is based on Support Vector Machines. The use of the nearest-neighbor paradigm is motivated by its use in a number of different pattern recognition fields, where it is difficult to produce a high-level generalization of a class of objects, but where neighborhood information is available [1, 6]. In particular, nearest-neighbor approaches have proven to be effective in outliers detection, and one-class classification tasks [3, 16]. Support Vector Machines are used because they are one of the most popular learning algorithm when dealing with high dimensional spaces as in CBIR [4, 17].

2.2.1 k-NN Relevance Feedback

In this work we resort to a technique proposed in [7] where a score is assigned to each image of a database according to its distance from the nearest image belonging to the target class, and the distance from the nearest image belonging to a different class. This score is further combined to a score related to the distance of the image from the region of relevant images. The combined score is computed as follows:

$$rel(\mathbf{I}) = \left(\frac{n/t}{1+n/t}\right) \cdot rel_{BQS}(\mathbf{I}) + \left(\frac{1}{1+n/t}\right) \cdot rel_{NN}(\mathbf{I}) \tag{1}$$

where n and t are the number of non-relevant images and the whole number of images retrieved after the latter iteration, respectively. The two terms rel_{NN} and rel_{BQS} are computed as follows:

$$rel_{NN}(\mathbf{I}) = \frac{\|\mathbf{I} - NN^{nr}(\mathbf{I})\|}{\|\mathbf{I} - NN^{r}(\mathbf{I})\| + \|\mathbf{I} - NN^{nr}(\mathbf{I})\|} \tag{2}$$

where $NN^{r}(\mathbf{I})$ and $NN^{nr}(\mathbf{I})$ denote the relevant and the non relevant Nearest Neighbor of $\mathbf{I}$, respectively, and $\|\cdot\|$ is the metric defined in the feature space at hand,

$$rel_{BQS}(\mathbf{I}) = \frac{1 - e^{1 - d_{BQS}(\mathbf{I}) \big/ \max_i d_{BQS}(\mathbf{I}_i)}}{1 - e} \tag{3}$$

where e is the *Euler's number*, i is the index of all images in the database and d_{BQS} is the distance of image $\mathbf{I}$ from a reference vector computed according to the Bayes decision theory (Bayes Query Shifting, BQS) [8]. If we are using F feature spaces, we have different scores $rel(\mathbf{I})$ for each f feature space. Thus the following combination is performed to obtain a "single" score:

$$rel(\mathbf{I}) = \sum_{f=1}^{F} w_f \cdot rel^{f}(\mathbf{I}) \tag{4}$$

where the w_f is the weight associated to the f-space. In this paper we are going to use two ways of computing the weights w_f. One approach to estimate the weights w_f is to take into account the minimum distance between all the pairs of relevant images, and the minimum distance between all the pairs of relevant and non-relevant images as follows

$$w_f = \frac{\sum_{i \in R} d^{f}_{min}(\mathbf{I}_i, R)}{\sum_{i \in R} d^{f}_{min}(\mathbf{I}_i, R) + \sum_{i \in R} d^{f}_{min}(\mathbf{I}_i, N)} \tag{5}$$

The other approach for estimating the weights w_f, is a modification of the previous one. Let us sort the images according to their distances from the query as measured by $rel(\mathbf{I})$, then their rank, from the closer to the farther, is considered. The weights are then computed by taking into account the relevant images and their "positions" in a f-space, and the sum of all the "positions" in all the feature spaces F as follows

$$w_f = \frac{\sum_{i=1}^{R} \frac{1}{pos_i^{f}}}{\sum_{k=1}^{F} \sum_{i=1}^{R} \frac{1}{pos_i^{k}}} \tag{6}$$

2.2.2 SVM based Relevance Feedback

Support Vector Machines are used to find a decision boundary in each feature space $f \in F$. The use of a SVM for this tasks is very useful because, in the case of image retrieval, we deal with high dimensional feature spaces. For each feature space f, a SVM is trained using the feedback given by the user. The results of the SVMs in terms of distances from the hyperplane of separation are then combined into to a relevance score through the Mean rule as follows

$$rel_{SVM}(\mathbf{I}) = \frac{1}{F} \sum_{f=1}^{F} rel_{SVM}^{f}(\mathbf{I}) \tag{7}$$

2.3 *Image Hunter's user interface*

The user interface is structured to provide just the functionalities that are strictly related with the user interaction (e.g., the list of relevant images found by the user). Image Hunter employs a web-based interface that can be viewed at the address *http://prag.diee.unica.it/amilab/WIH*. This version is a web application built for the Apache Tomcat web container by using a mixure of JSP and java Servlet. The graphic interface is based on the jQuery framework, and has been tested for the Mozilla Firefox and Google Chrome browsers. When the web container is launched, a servlet checks if the folder of the collection contains an updated Lucene index; if not, the index is updated. Afterward, the index is loaded by the web application and used for all the sessions opened by the remote clients (see Figure 2). The Image

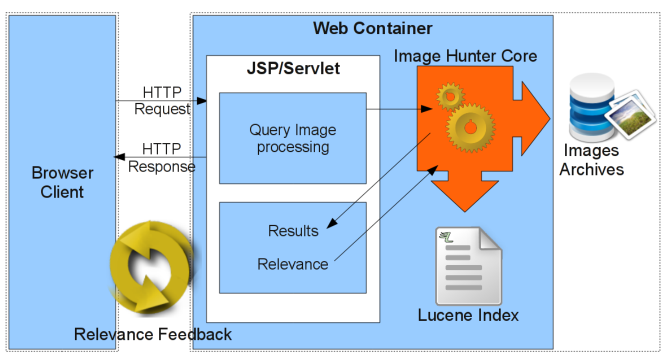

Fig. 2 Web Application Architecture

Hunter homepage let the user choose the picture from which starting the search. The picture can be chosen either within those of the proposed galleries or among the images from the user hard disk (see Figure 3). Each query is managed by a servlet that queries the Image Hunter engine and displays the 23 most similar im-

Fig. 3 Web Application

ages according to the mechanisms reported in Section 2.2. The choice of the number of images displayed to the user on the one hand takes into account the needs of the page layout and, on the other hand, is oriented to maintain high the user attention. In order to make more intuitive and easy the features of the application, the graphical interface has been designed relying on the *Drag and Drop* approach (see Figure 4). From the result page the user can drag the images that her deems relevant to her search in a special boxcart, and then submit the feedback. Then the feedback is processed by the system, and a new set of images is proposed to the user. The user can then perform another feedback round.

In order to make the system more flexible for skilled users, the *Settings* page allows choosing the low-level feature used to describe the image content. In particular, it is possible to select between 7 color based descriptors that are: *Scalable Color*, a color histogram extracted from the HSV color space; *Color Layout*, that characterizes the spatial distribution of colors; *RGB-Histogram* and *HSV-Histogram*, based on RGB and HSV components of the image respectively; *Fuzzy Color*, that considers the color similarity between the pixel of the image; *JPEG Histogram*, a JPEG coefficient histogram, and *ABIF32* obtained rescaling the images to 32x32 size and returning a color histogram extracted from the RGB color space. It is also possible choosing between different texture and shape features that are: *EDGE Histogram*, that captures the spatial distribution of edges; *Tamura*, that captures different characteristic of the images like coarseness, contrast, directionality, regularity, roughness, and *Gabor* that allows the edge detection. In addition, it is possible to use two descriptors that merge color and texture characteristics: *CEDD* (Color and Edge Directivity Descriptor), and *FCTH* (Fuzzy Color and Texture Histogram).

One of Image Hunter's greatest strengths is its flexibility, as it is possible to add any other image descriptor. The choice of the above mentioned set is due to the "real time" nature of the system. In fact even if some local features such as SIFT or SURF could improve the retrieval performances for some particular kind of searches, on the other hand they are more time expensive in the evaluation of the similarity between images.

In addition, in the *Settings* page the user can select the Relevance Feedback technique to be used, and the dataset to explore.

Fig. 4 Results and Relevance Feedback

3 Experiments

3.1 Dataset Setup

In the experimental evaluation of Image Hunter we performed both a full automatic test by using the MIRFLICKR-25000 collection [10] and a *User Experience* test using a subset of 53279 unlabelled images extracted from the *Sardegna Digital Library*[3]. MIRFLICKR-25000 consists of 25000 images tagged by the user of the social photography site Flickr. The average number of tags per image is 8.94. In the collection there are 1386 tags which occur in at least 20 images. Moreover, for a limited number of images, some manual annotations is also available (24 annotations in the collection considered for this experiment). In these experiments we used all the features embedded with the system that have been listed in Section 2.3.

In the automatic test, we analyzed all the tags of the collection by a semantic point of view, and fused the tags with the annotations in a tag verification process. This process was performed to keep only the tags which occur in at least 100 images, so that the single tags/concepts are adequately represented in the dataset used in the evaluation experiments. This process of fusing and discarding tags brought us to keep 24718 images and 69 tags, with an average number of tags per image of 4.19. Thus, as “starting” query images, we chose 1294 of them from the refined collection. These query images have a number of tags per image that varies from 3 to 10 (i.e., the single image can represent different meanings) , with an average number of tags per image equal to 4.69 (thus very similar to the value in all the collection). For each one of the 1294 query image, a relevance feedback experiments had been performed by using all the tags as a target, i.e., given a query image, we considered, one at a time each single tag as target of the retrieval process to be refined through the relevance feedback. Thus, each query image has been used as starting example

[3] http://www.sardegnadigitallibrary.it

for different retrieval tasks. In this way, 6070 retrieval tasks were performed for each relevance feedback technique implemented inside *Image Hunter*.

Each automatic experiment consists of 10 iterations: the first one is based on a nearest neighbor search on all the feature spaces, and the other 9 iterations are based on one of the relevance feedback techniques described above. At each iteration we simulated the feedback from the user on 20 images.

The *User Experience* test has been performed by 52 users that were asked to perform one or more searches by choosing as query one out of 32 images (See Figure 5) that we selected so that they exhibited different subjects, different colors and shapes. The users can choose to perform any number of consecutive iterations to refine the search. On average, each of the 32 queries has been used 6.75 times and the users performed an average of 5 iterations. At each iteration $n = 23$ images are shown to the user for marking the feedback.

Fig. 5 User Experience queries

3.2 Performance measurements

The performance of the experiments will be assessed using the *Precision* and a modified definition of the Recall, that we named "user perceived" *Recall*.

The *Precision* is a measure that captures how many relevant images are found within the images that are "shown" to a user, and it is computed as follows:

$$p = \frac{A(q) \cap R(q)}{A(q)} \tag{8}$$

where $A(q)$ is the ensemble of images retrieved by using the query q, while $R(q)$ is the ensemble of images that in the collection are relevant according to the query q.

The *Recall* measures how many relevant images are found among the set of images in the collection that have the same tag/concept:

$$r = \frac{A(q) \cap R(q)}{R(q)} \tag{9}$$

In this way we compute the percentage of relevant images with respect to the totality of relevant images in the collection. This measure has a disadvantage: if the total number of relevant images in a collection for a given tag is greater than the number of images shown by the system, the measure is going to be always less than 100% even if all the images shown to the user are relevant. Thus, this measure doesn't represent the perception, in term of performance, that a real user will have on the system. In addition, each class contains a different number of images, and therefore the denominator of Equation 9 differs from one class to another even in one order of magnitude, and it can completely distort the average performance. For these reasons, we propose to use a modification of the recall measure namely, the "user perceived" *Recall*. This is a recall measure takes into account just the maximum number of relevant images that can be shown to the user according to the number of iterations, and the number of images displayed per iteration, and it is computed as follows

$$r_p = \frac{A(q) \cap R(q)}{R^*(q)} \quad , \quad R^*(q) = \begin{cases} R(q) & \text{, if } |R(q)| \leq n \cdot i \\ n \cdot i & \text{, otherwise} \end{cases}$$

where $A(q)$ is the number of images retrieved by using the query q up to the iteration i, $R(q)$ is the number of relevant images in the dataset (for the query target), $|\cdot|$ indicates the cardinality of the set, and n is the number of images shown to the user per iteration.

In an unlabelled dataset it is more difficult to define the set of similar images, so in the *User Experience* test for each query image we have formed the set of relevant images by considering the images marked as Relevant by at least the 50% of users. Indicating as $\widehat{R}(q)$ this ensemble of images retrieved by using the query q the *Recall* will be:

$$r = \frac{A(q) \cap \widehat{R}(q)}{\widehat{R}(q)}. \tag{10}$$

It is worth to note that $|\widehat{R}(q)| \leq n \cdot i$, so the *Recall* and the "user perceived" *Recall* agree.

3.3 Experimental results

In the automatic test we compared the performance of all the relevance feedback techniques described in the previous section, i.e., the k-NN based on Equation (5) (*NN* in the tables), the k-NN based on Equation (6) (*PR* in the table), and the SVM, with the performance attained by simply *browsing* the image collection. In the *User Experience* test, we just used the NN relevance feedback mechanism. The term *browsing* indicates nothing more than showing the user the n images nearest to the query with no feedback. The aim of comparing relevance feedback with *browsing* is to show the benefits of relevance feedback. To put it simple: can a relevance feedback approach retrieve more relevant images than simply *browsing* the collection by sorting the images according to the visual similarity with the query?

The average results in terms of *Precision*, and "user perceived" *Recall* obtained in the automatic test are presented in Table 1. The results show that, as the number of iterations increase, the performance of the relevance feedback methods increase, as well as the difference in performance with the *browsing*. From these analysis it turns out that the behavior of the two k-NN methods are quite similar, while the SVM exhibits the biggest increasing performance power. To evaluate the *User Experience*,

Precision

it.	0	1	2	3	4	5	6	7	8	9
SVM	29.6	28.0	**30.5**	**33.1**	**35.2**	**36.9**	**38.3**	**39.4**	**40.4**	**41.1**
NN	29.6	**28.7**	29.4	30.0	30.6	31.1	31.5	31.8	32.1	32.4
PR	29.6	28.5	29.3	30.1	30.7	31.1	31.5	31.9	32.1	32.3
browsing	29.6	28.6	28.1	28.0	27.7	27.5	27.3	27.2	27.1	26.9

Recall

it.	0	1	2	3	4	5	6	7	8	9
SVM	3.0	5.6	**9.2**	**13.2**	**17.6**	**22.2**	**26.9**	**31.6**	**36.4**	**41.2**
NN	3.0	**5.8**	8.8	12.0	15.3	18.7	22.0	25.5	28.9	32.3
PR	3.0	5.7	8.8	12.0	15.4	18.7	22.1	25.5	28.9	32.3
browsing	3.0	5.7	8.5	11.2	13.9	16.6	19.2	21.8	24.4	27.0

Table 1 Precision and "User perceived" *Recall* in the MIRFlickr experiments.

we show the values of the recall measure: Table 2 reports the obtained results. The performance of the RF technique w.r.t. the browsing shows as the user interaction permits a very big improvement of the performance and as the system learn how to find images that fulfil the user's desires. We observed that the users tend to label as "Relevant" less and less images after few iterations, especially if she is satisfied with the previous results, because labelling the images is an annoying task. As a consequence, the reported values of the recall can be considered as a lower bound of the true performances, as they take into account just the images actually labelled by the user.

Recall

it.	0	1	2	3	4
NN	34,4	**59,6**	**74,2**	**77,4**	**81,8**
browsing	**41,2**	48,8	53,1	55,6	55,6

Table 2 Recall in the SDL experiments.

4 Conclusions

In this paper we presented Image Hunter, a tool that exploits the potentiality of Relevance Feedback to improve the performance of Content Based Image Retrieval.

Unlike other proposed tools, Image Hunter is a full content based image retrieval system in which the user's feedback is integrated in the core of the application, and permits a dynamical adaptation of the queries driven by the user. The proposed results obtained both in a full automatic test, and in a user test show how the integration of the relevance feedback improves significantly the performance of the image retrieval system making the search more effective w.r.t. the web browsing.

References

1. Aha, D.W., Kibler, D., Albert, M.K.: Instance-based learning algorithms. Machine Learning **6**(1), 37–66 (1991)
2. Barthel, K.U.: Improved image retrieval using automatic image sorting and semi-automatic generation of image semantics. In: WIAMIS '08: Proceedings of the 2008 Ninth International Workshop on Image Analysis for Multimedia Interactive Services, pp. 227–230. IEEE Computer Society, Washington, DC, USA (2008).
3. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: LOF: Identifying density-based local outliers. In: W. Chen, J.F. Naughton, P.A. Bernstein (eds.) SIGMOD Conference, pp. 93–104. ACM (2000).
4. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press (2000)
5. Datta, R., Joshi, D., Li, J., Wang, J.Z.: Image retrieval: Ideas, influences, and trends of the new age. ACM Computing Surveys **40**(2), 1–60 (2008).
6. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. John Wiley and Sons, Inc., New York (2001)
7. Giacinto, G.: A nearest-neighbor approach to relevance feedback in content based image retrieval. In: CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval, pp. 456–463. ACM, New York, NY, USA (2007).
8. Giacinto, G., Roli, F.: Bayesian relevance feedback for content-based image retrieval. Pattern Recognition **37**(7), 1499–1508 (2004).
9. Huang, T., Dagli, C., Rajaram, S., Chang, E., Mandel, M., Poliner, G., Ellis, D.: Active learning for interactive multimedia retrieval. Proceedings of the IEEE **96**(4), 648 –667 (2008).
10. Huiskes, M.J., Lew, M.S.: The MIR flickr retrieval evaluation. In: M.S. Lew, A.D. Bimbo, E.M. Bakker (eds.) Multimedia Information Retrieval, pp. 39–43. ACM (2008).
11. Lew, M.S., Sebe, N., Djeraba, C., Jain, R.: Content-based multimedia information retrieval: State of the art and challenges. ACM Trans. Multimedia Comput. Commun. Appl. **2**(1), 1–19 (2006).
12. Lux, M., Chatzichristofis, S.A.: Lire: lucene image retrieval: an extensible java cbir library. In: MM '08: Proceeding of the 16th ACM international conference on Multimedia, pp. 1085–1088. ACM, New York, NY, USA (2008).
13. Rui, Y., Huang, T.S.: Relevance feedback techniques in image retrieval. In: Lew M.S. (ed.): Principles of Visual Information Retrieval, pp. 219–258. Springer-Verlag, London (2001)
14. Segarra, F.M., Leiva, L.A., Paredes, R.: A relevant image search engine with late fusion: mixing the roles of textual and visual descriptors. In: P. Pu, M.J. Pazzani, E. André, D. Riecken (eds.) IUI, pp. 455–456. ACM (2011)
15. Smeulders, A.W.M., Worring, M., Santini, S., Gupta, A., Jain, R.: Content-based image retrieval at the end of the early years. IEEE Trans. Pattern Anal. Mach. Intell. **22**(12), 1349–1380 (2000).
16. Tax, D.M.: One-class classification. Ph.D. thesis, Delft University of Technology, Delft, The Netherlands (2001).
17. Tong, S., Chang, E.: Support vector machine active learning for image retrieval. In: Proc. of the 9th ACM Intl Conf. on Multimedia, pp. 107–118 (2001).
18. Zhou, X.S., Huang, T.S.: Relevance feedback in image retrieval: A comprehensive review. Multimedia Syst. **8**(6), 536–544 (2003).