IMAGE RETRIEVAL OVER NETWORKS: ANT ALGORITHM FOR LONG TERM ACTIVE LEARNING

David Picard[1], *Arnaud Revel*[2]

[1]ETIS, CNRS, ENSEA, Univ Cergy-Pontoise,
[2]ENSEA
F-95000 Cergy-Pontoise

Matthieu Cord

LIP6 UPMC
Paris 6
104 avenue du Président Kennedy
75016 Paris FRANCE

ABSTRACT

Image retrieval over a network is the focus of this paper, as being a major challenge of content based image retrieval. We present a system that gathers feedbacks given by the users in order to learn the location of the searched images. As a result, the active learning of a content based relevance function is enhanced by the selection of hosts containing relevant examples. We achieve a long term merging of feedbacks over many sessions, without any knowledge about the category being searched at any session. Our system is based on mobile agents crawling the network in an ant-like behavior. Markers are used to learn a routing of the agent leading them to the relevant images. The benefits of the ant-like agents system is a natural parallelization of the processing as well as a distributed approach to the routing learning. We made experiments on the trecvid'05 key-frame dataset showing that the location of the categories were efficiently learned. Furthermore, the long-term learning of categories improves the interaction by reducing the number of labels needed during the interaction to obtain satisfying results.

1. INTRODUCTION

As multimedia devices (such as mobile phones, digital cameras, etc.) are becoming very usual, huge collections of digital images are available today. Finding images belonging to a specific category in these ever growing collections is a difficult task since searching within by hand has become impossible. Content Based Image Retrieval (CBIR) has been successfully proposed to answer this problem [1]. The main idea is to build a description based on the images content, and to find similarities between descriptions [2]. The problem of such techniques is the well known *semantic gap* between the numerical values attached to images and the semantical concepts they belong to. In order to reduce the gap, machine learning techniques have been successfully adapted to train a similarity function in interaction with the user (using her labeling of the results) leading to the so called "relevance feedback" [3, 4]. The best improvement has been done with the introduction of active learning, which aims at proposing for labeling the image that will at most enhance the similarity function when added to the training set [5].

With the expansion of networks such as the Internet, peer-to-peer networks or even personal networks, image retrieval has become a difficult task. As images are split into many collections over the web, the problem of CBIR is not only to find the most relevant images, but also to find the localization of relevant collections. The major part of CBIR computation being dedicated to the processing of the image descriptors, the fact that images are distributed over many sources should be more an advantage than a drawback since it means a possible paralleling. Moreover, we can assume that collections may be coherent in semantics, in other words, that images belonging to a specific concept may be encountered in only few well localized collections. For instance, a site can mostly contain touristic photographs (landscapes, buildings, etc.), while an other can mainly contain photographs of manufactured objects, and a third can only contain old paintings. Although CBIR in a distributed context has been noted as an interesting improvement [6], it has been, to our knowledge, the focus of a few works. Chen presented a system for image retrieval in p2p network in CBMI'07 [7]. In their system, the links between peers of the network are optimized in order to propagate the query to relevant hosts. We have proposed in a previous work [8] to learn the location of relevant images from the interaction with the user. We carried out a smart cooperation between the interactive CBIR and a localization learning in a global architecture based on mobile agents.

However, all the labels gathered during the interaction are forgotten at the end of the session of a classical CBIR system. The major challenge of a widely used CBIR system is to re-use these labels for later sessions in order to benefit from the previous interactive learning [9]. In a single-user CBIR system, the resulting long-term learning is possible but very slow due to the few labels available. The real interest is in our distributed context since we can gather labels over sessions from many users in parallel as the network is shared between users. In that sense, the knowledge given to the system through the relevance feedback can be gathered from all users. In this paper, we present a generalization for long-term optimization of our previous CBIR over networks system. The localization of the categories are learned over several sessions, enabling a routing of the mobile agents specific to the searched concept. We assume that the categories one might search of are well localized on the networks. Such specialized collections are already available, as for example collections of medical images or aerial images. Based on this assumption, we will learn a category dependent routing of the networks in order to retrieve images from relevant locations.

In the next section, an overview of our system is exposed. The section 3 contains the description of the routing learning algorithm during a session. Section 4 describes the long-term optimization. Finally, we present and discuss the experiments and results we obtained using our system on the trecvid2005 key-frame dataset[1] in section 5.

2. RETRIEVAL SCHEME

Our system is based on mobile agent technology. A mobile agent is an autonomous computer software with the ability to migrate from one computer to another and to continue its execution there. There are good reasons for using mobile agents in the distributed CBIR context, such as the reduction of the network load (the processing code of the agent being very small in comparison to the feature vector indexes) and the massive paralleling of the computation [10].

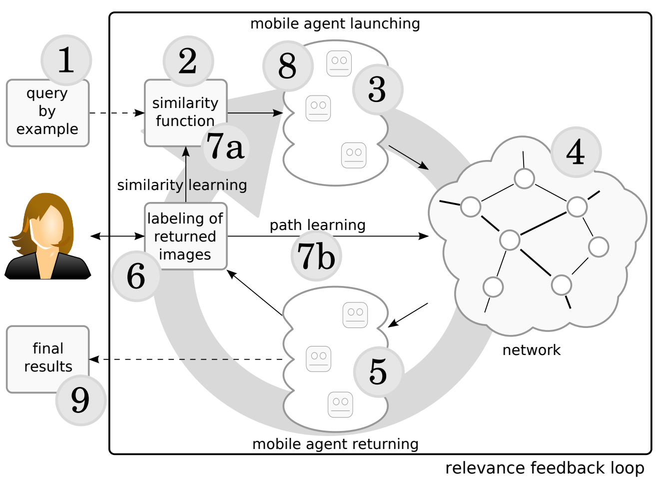

Fig. 1. Functional description of our system showing the user in interaction with the relevance feedback loop (launching of agents, retrieval, display and labeling).

As described in Fig. 1, the user starts his query by giving an example or a set of examples to an interface (1). A similarity function based on these examples is built (2). Mobile agents are then launched with a copy of this similarity function (3). Every host of the network contains an agent platform in order to be able to receive and execute incoming mobile agents. The agent movements are influenced by markers (a numerical value locally stored on the host) following an ant-like behavior [11, 12], as described in section 3

On each platform, an agent indexing the local images is run, and retrieves the relevant images for the incoming agents. As soon as they receive the answer of the index agent, the mobile agents return to the user's computer (5) and the results are displayed on the interface (6). The user can label these results (1: *relevant*, -1: *irrelevant*), and the similarity function is updated consequently (7a) as well as the relevant paths of the network (7b). As the similarity function we use is based on SVM analysis [13], the update only consists in adding the results and their labels to the training set and to train a new SVM.

Mobile agents are then relaunched with the improved similarity function. To tackle the *semantic gap* problem, an interactive loop consisting in several launching of mobile agents and labeling of the results is set (8). At the end of the interaction, mobile agents are launched for a very last time in order to retrieve the best results from each host (9). The number of retrieved images is proportional to the level of the markers leading to this host, assuring that most of the best retrieved images are provided by relevant hosts.

3. ROUTING LEARNING

We give here a rapid overview of our learning algorithm described in [8]. During a session, the similarity function is learned as well as a routing of the agents leading them to host containing relevant images. This routing is done by the ant-like behavior of the agents. While moving from one host to another, agents are influenced by markers regarding the following rule :

$$P_i = \frac{ph_i}{\sum_{k \in S} ph_k} \quad (1)$$

Where P_i is the probability of an agent to move to host i, ph_i being the value of marker of the host i, and S the possible destinations. We use an ant-like algorithm [14, 15] following the model of the ethologist J. L. Deneubourg [16] to reinforce the markers. The markers act like pheromones for ants, in the sense that ants tend to follow the path containing the highest quantity of pheromone while searching for food. Once they have found food, they go back to the nest laying a trail of pheromone on the path they followed. This trail will influence later ants so as to reflect the direction of the found sources. In our system, the level of markers shall reflect the relevance of the host regarding the searched category. Each time an agent moves towards a host, the selected marker is decreased as follows:

[1] see http://www-nlpir.nist.gov/projects/tv2005/

$$\Delta ph_i = -\alpha \cdot ph_i \quad (2)$$

This rule models the evaporation of the pheromones for real ants. The paths that are not positively reinforced will have their markers level very low due to this rule. Each time the user labels an image, the selected markers on the pathway taken by the agents to retrieve this image are increased:

$$\Delta ph_i = +\gamma \cdot u \quad (3)$$

With u being 1 if the label is positive, 0 otherwise. This rule models the deposit of pheromone by real ants once they have found food. Hosts containing relevant images will be reinforced by this rule, and thus, their markers will have a high level. Using these rules, the estimation of the marker $\hat{ph_i}$ is dependant on the estimation of $\hat{u}$ the labels given by the user:

$$\hat{ph_i} = \frac{\gamma \cdot \hat{u}}{\alpha} \quad (4)$$

Thus the higher levels of marker will be obtained for the hosts that gave the greatest number of positive labeling, leading to a routing associated with the session's category.

We ran experiments on the Trecvid'05 keyframes dataset to evaluate the quality of the routing learning. In a network consisting of four hosts plus the computer of the user, we hosted the searched category on the fourth host. The remaining images were equally distributed on the four hosts. We then ran a session consisting of 100 labels given by the user. The Fig. 2 shows the mean probability to visit each of the four hosts at the end of the session.

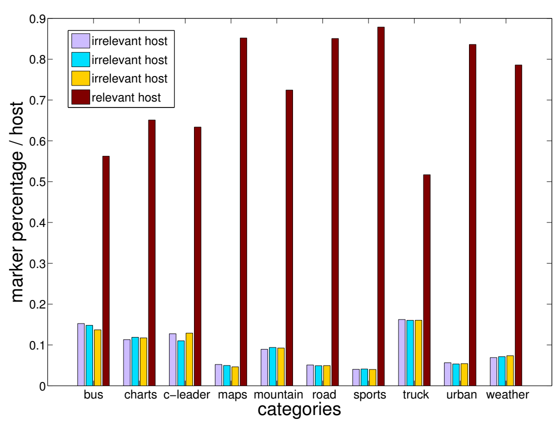

Fig. 2. Results of the routing learning system : final probability score P_i of an agent to move to each host at the end of a retrieval session depending on the category being searched. In our experiment, fourth host of the four destinations available contained the relevant category. At the end of a session, its probability P_4 of being visited is the highest.

As one can see, the fourth host has always the highest probability of being visited by the agents, showing that the learning of the relevant destination is succesfull. Moreover, some categories were more difficult to localize with the ant-algorithm. This phenomenon can be explained by the mean reinforcement $\hat{u}$ being lower for difficult categories (where more negative examples are found by the active strategy) than for easy categories (where more positive examples are found by the active strategy).

4. LONG TERM MERGING OF SESSIONS

The main contribution of this paper is the generalization of our previous system to long term leaning. We propose an architecture to merge all the information registered during past retrieval sessions. After several sessions, we dispose of several category related routings as explained in section 3. In order to re-use the routings in sessions concerning an already learned category, we put on each host i of the network N markers $\{ph_{i,j}\}_{1 \leq j \leq N}$, each of them being related to a specific category. Les us denote $\mathbf{Ph}_i$ the vector concatenating the $ph_{i,j}$ available on the host i.

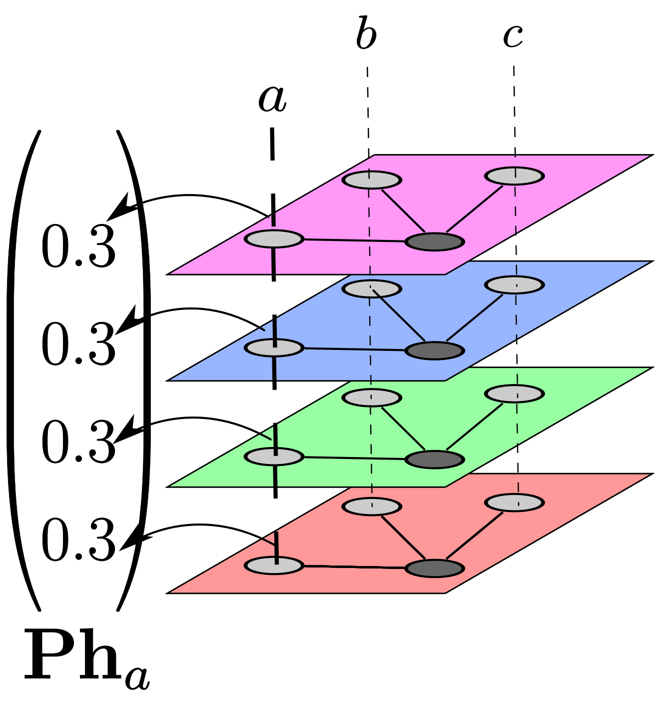

Fig. 3. Long-term system architecture before learning: in this example, the user's computer is linked to three target hosts a, b and c. There are four planes available to catch the distribution of the categories. The probabilities to visit a host are uniform for each plane. For instance, the probability to visit a is $1/3$ on all of the four planes.

The use of the markers $\{\mathbf{Ph}_i[j]\}$, j being fixed, for all host i of the network leads to a routing relevant for the as-

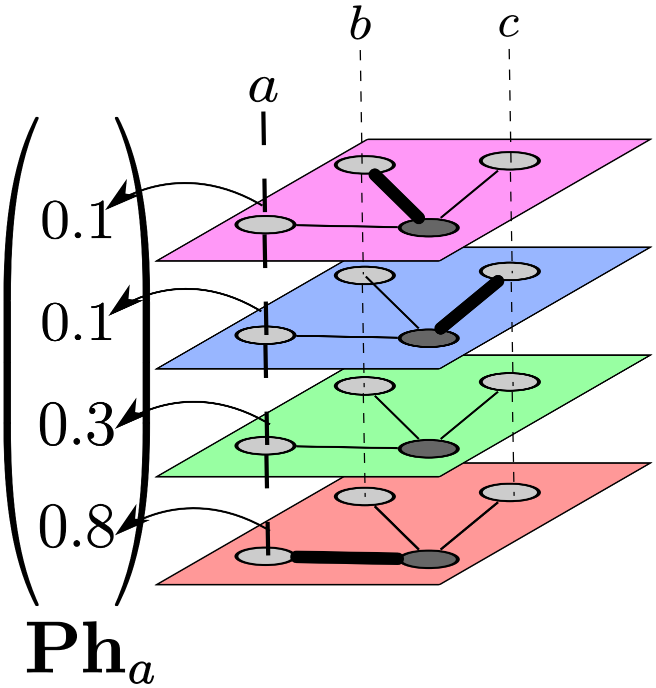

Fig. 4. Example of the long-term system after learning: Some of the planes have specialized into one category. For instance, the fourth plane has specialized to the category contained on host a and therefore its probability of being visited using this plane is 0.8.

sociated category. Let us denote a *plane* such a routing (see Fig. 3 and 4). The association between a plane and the related category is not known, since the system is not aware of which category is being searched at any session (users do not give a label to the session, for instance). The goal of a long term learning is to find, based on the interaction with the user, if the current category has already been searched, and if so to reuse the information obtained from previous corresponding sessions. In order to benefit from these previously learned routings, we build a function ψ selecting the marker $ph_i[j]$ corresponding to the searched category. The goal of ψ is to select a plane related to the searched category. This plane is used by the agents to move. The function ψ is obtained from a vector Ψ containing 1 for the relevant plane, and 0 otherwise :

$$\psi(\mathbf{Ph}_i) = \Psi^\top \cdot \mathbf{Ph}_i \tag{5}$$

As we do not have any *a priori* about the category being currently searched and about the categories associated with the available planes, we build the vector Ψ in interaction with the user. We associate with each plane j a probability W_j of being related to the currently searched category. We sample the plane to be used by an agent regarding these propabilities using a multinomial law $\mathcal{M}(1; W_1, \ldots, W_N)$. Ψ is obtained as the projection on the selected plane:

$$\Psi = \{\delta_{j,m}\}_{1\leq j\leq N} \quad , \quad m \textit{ being the selected plane} \tag{6}$$

Each time the user labels an image, the W_j corresponding to the markers that have been used to retrieve this image evolves regarding the following rule:

$$\Delta W_j = \varepsilon(u - W_j) \tag{7}$$

Where $u = 1$ if the label is positive, 0 otherwise. Thus, planes that gave a lot of positive labels will have a higher weight, which means a higher chance of being selected.

Ψ is reset each time an agent is launched. A new vector is sampled thanks to $\mathcal{M}(1, W_1, \ldots, W_N)$ with the updated weights. All the dimensions are explored until the weights converge to the relevant plane.

As both the function ψ and the markers level are learned at the same time (using the same reinforcement u given by the user), the dynamics of the markers evolution is set slower than the one of ψ. Consequently, a set of marker is chosen (by convergence of the weights) before the markers are evolved.

5. EXPERIMENTS

We used the trecvid2005 key-frame dataset to test the influence of our category dependent routing on the retrieval. We put the three categories we tested (namely *airplane*, *explosion-fire* and *maps*) on three different hosts, and added about 4000 randomly chosen images from the category *entertainment* to each host (simulating the various content a real host contains). These were the possible destinations of our mobile agents. We ran one hundred of retrieval sessions which consist of launching of agents and displaying of results until 100 labels where obtained. At any session, the searched category was randomly chosen within the three hosted categories.

As shown on Fig. 5, each host had some of the markers specializing in it. For instance, the third host has been routed by the second and last planes, which means that a function ψ choosing one of these two planes will lead the agents to host 3.

To illustrate the evolution of the $\mathbf{Ph}_i[j]$ during the learning, Fig. 6 shows the distribution of the markers for all sessions. In this diagram, the marker coordinates (x_j, y_j) are a projection of their values on each of the 3 host on a 2D space at each of their updates:

$$(x_j, y_j) = (\sum_{i=0}^{2} \mathbf{Ph}_i[j] \cdot \cos(\frac{i \cdot 2\pi}{3}), \sum_{i=0}^{2} \mathbf{Ph}_i[j] \cdot \sin(\frac{i \cdot 2\pi}{3})) \tag{8}$$

The hosts where a marker has the highest probability are sectors delimited by lines. Some of the markers specialized very quickly and remained specialized for a single category during all the sessions (like the first plane), while others did

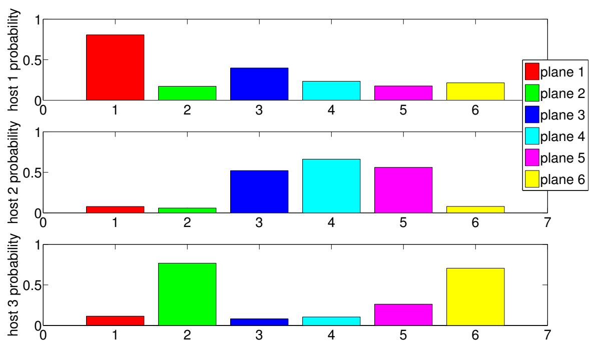

Fig. 5. Relative values of each marker on the three hosts. Each of the six planes has been specialized for one host. For example, the first plane has a higher probability to lead to the first host, while the second plane has a higher probability to lead to the third host.

move from one category to another (like the third plane). At least one plane keeps the relevant routing for each category (the first plane for the first category, the fifth plane for the second category, and the last plane for the third category) during all the sessions. In that sense, the system is stable.

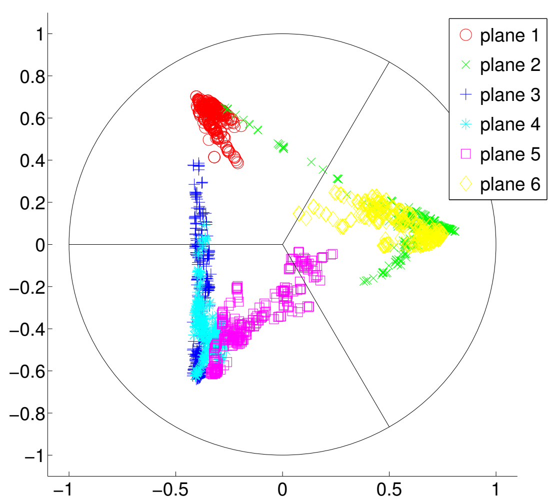

Fig. 6. Distribution of markers on the three hosts for all session. Some markers (for example markers of the first plane) stay in the zone a one host for almost all sessions, consequently routing the agents to this host.

The markers that were used for each category are shown on Fig. 7. As we can see, all categories used a subset of the markers available, meaning that the markers did specialized for a category. We can clearly see the correlation between this specialization and the Fig. 5. The markers used for a session concerning a category where exactly those leading to the host which contained it.

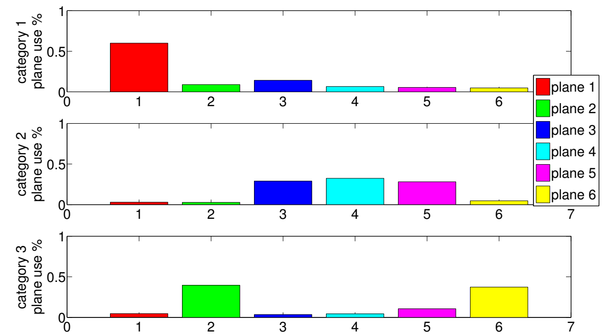

Fig. 7. Percentage of markers that were used for each category. Markers have been at most used for the category towards which destination they routed. For example, for the third category, the most used markers where the second and the sixth, which are leading to the host containing the third category as seen on Fig. 5

Fig. 8 shows the recall obtained for 500 images retrieved for each category. In order to see the benefit of our distributed over classical CBIR system, we ran the same experiments with all the images localized on a single system. In this *centralized setup*, there is no network, and thus no routing learning. Our system performs slightly better, although it is not very pronounced. As discussed in the next figure, the maximum performance is obtained for all categories with both systems, which can explain the small gain obtained after the long-term optimization. The main advantage our distributed system over the classical CBIR setup is the time to perform the retrieval, which was about four time less.

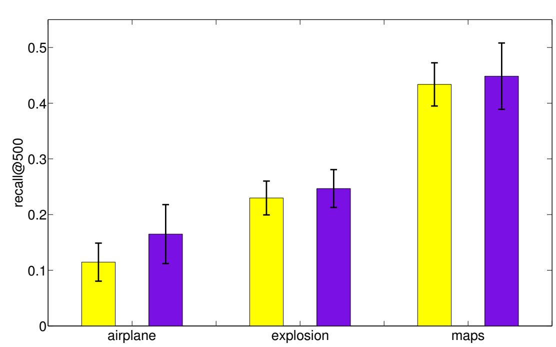

Fig. 8. Recall for each category for 500 images retrieved, compared to a classical CBIR system. The distributed system performs a little better on more difficult categories as *airplane* or *explosion-fire*.

The improvement of the mean average precision (*MAP*) due to long-term learning is shown on Fig. 9, Fig. 10 and Fig. 11. As one can see, the long-term optimization leads to an improvement of the *MAP* between 5% and 10%. For a difficult category like *airplane*, the gain is about 5%, whereas for an easy category like *explosion-fire*, the gain is about 8%. The main consequence is that an equivalent *MAP* can be obtained with fewer labels after the long-term optimization. For instance, to obtain the same *MAP* as with 100 labels before long-term optimization, about 40 were needed for the *maps* category, about 50 for the *explosion* category and about 70 for the *airplane* category after the long-term optimization.

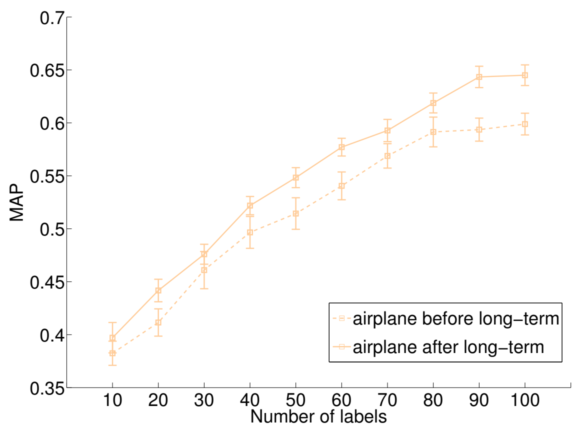

Fig. 9. MAP for the category *airplane* before and after learning. With the long-term optimization, the gain is about 5% and the speed-up is about 30 labels.

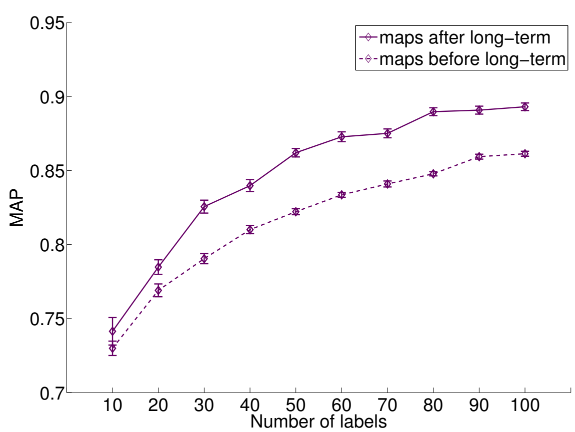

Fig. 10. MAP for the category *maps* before and after learning. With the long-term optimization, the gain is 5% and the speed-up is about 50 labels.

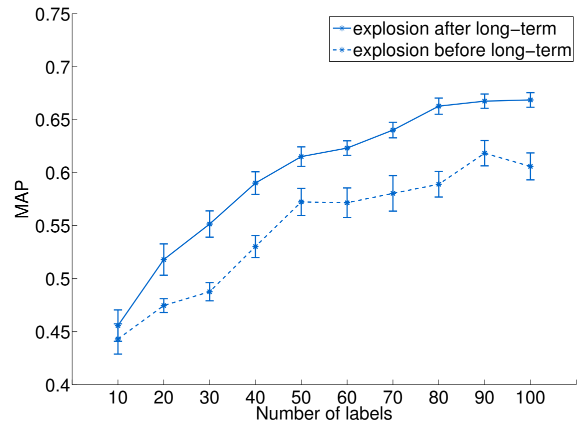

Fig. 11. MAP for the category *explosion-fire* before and after learning. With the long-term optimization, the gain is about 8% and the speed-up is about 60 labels.

6. CONCLUSION

In this paper we have presented a CBIR system based on mobile agent technology with an ant-like behavior. Agents crawl the network in search of images using a relevance function learned in interaction with the user. The locations from where the images are retrieved depend on a set of markers. The higher the level of a marker, the higher the probability of the agents to move to this host. Markers are reinforced unsing the labels given by the user. Host containing positively labeled images have their markers increased, whereas host containing more negatively labeled images have their markers level very low. We use several markers on each host, each being associated to a category. Thus, agents using a specific markers set will be routed towards the host containing the associated images. The association between these markers and categories is learned in interaction with the user, resulting into a category dependent routing. Our system carries out a smart cooperation between this routing and the active learning of the similarity function used for the retrieval, leading to an improvement of the recall.

While markers can be naturally shared between users, our system builds an user oriented semantic map of the network that can be used efficiently to improve the retrieval. As shown on the experiments, the association learned between a set of markers and a category improves the retrieval. While the recall gain is little compared to a classical CBIR system, the speed up in learning obtained through the long-term is high. For instance, for the category *maps*, only 40 labels were needed to obtained about the same MAP as with 100 labels without the long-term optimization.

7. REFERENCES

[1] R.C. Veltkamp, "Content-based image retrieval system: A survey," Tech. Rep., University of Utrecht, 2002.

[2] W. Niblack, R. Barber, W. Equitz, M. Flickner, E.H. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin, "The QBIC project: Querying images by content, using color, texture, and shape," in *Storage and Retrieval for Image and Video Databases (SPIE)*, February 1993, pp. 173–187.

[3] M.E.J. Wood, N.W. Campbell, and B.T. Thomas, "Iterative refinement by relevance feedback in content-based digital image retrieval," in *ACM Multimedia 98*, Bristol, UK, September 1998, pp. 13–20.

[4] T.S. Huang and X.S. Zhou, "Image retrieval with relevance feedback: From heuristic weight adjustment to optimal learning methods," in *International Conference in Image Processing (ICIP'01)*, Thessaloniki, Greece, October 2001, vol. 3, pp. 2–5.

[5] Simon Tong and Daphne Koller, "Support vector machine active learning with applications to text classification.," *Journal of Machine Learning Research*, vol. 2, pp. 45–66, 2001.

[6] A.W.M Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1349–1380, December 2000.

[7] Jiann-Jone Chen, Chia-Jung Hu, and Chun-Ron Su, "Scalable image retrieval with optimal configuration for p2p network database," *Content-Based Multimedia Indexing, 2007. CBMI '07. International Workshop on*, pp. 236–243, 25-27 June 2007.

[8] David Picard, M. Cord, and Arnaud Revel, "Cbir in distributed databases using a multi-agent system," in *IEEE International Conference on Image Processing (ICIP'06)*, Atlanta, GA, USA, October 2006.

[9] M. Cord and P.-H. Gosselin, "Image retrieval using long-term semantic learning," in *IEEE International Conference on Image Processing*, Atlanta, GA, USA, oct. 2006, IEEE.

[10] Danny B. Lange and Mitsuru Oshima, "Seven good reasons for mobile agents," *Commun. ACM*, vol. 42, no. 3, pp. 88–89, 1999.

[11] A. Revel, "Web-agents inspired by ethology: a population of "ant"-like agents to help finding user-oriented information.," in *IEEE WIC'2003 : International Conference on Web Intelligence.*, Halifax, Canada, October 2003, IEEE, pp. 482–485, IEEE Computer Society.

[12] Arnaud Revel, "From robots to web-agents: Building cognitive software agents for web-information retrieval by taking inspiration from experience in robotics," in *ACM International conference on Web Intelligence*, Université Technologique de Compiègne, Compiègne, France, Septembre 2005.

[13] Corinna Cortes and Vladimir Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[14] M.Dorigo, V. Maniezzo, and A. Colorni, "The ant system: Optimization by a colony of cooperating agents.," *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, vol. 1, no. 26, pp. 29–41, 1996.

[15] E. Bonabeau, M. Dorigo, and G. Theraulaz, "Inspiration for optimization from social insect behaviour," *Nature*, vol. 406, pp. 39–42, 6 July 2000.

[16] J.L. Deneubourg, S. Goss, N.Franks, A. Sendova-Franks, C. Detrain, and L. Chrétien, "The dynamics of collective sorting: Robot-like ants and ant-like robots," in *From Animals to Animats: Proc. First Int. Conference on Simulation of Adaptive Behavior*, J-A. Meyer and S.W. Wilson, Eds., Paris, France, 1990, pp. 356–363.