An Application of Swarm Intelligence to Distributed Image Retrieval

David Picard, Arnaud Revel, Matthieu Cord

An Application of Swarm Intelligence to Distributed Image Retrieval

David Picard[*,a], Arnaud Revel[b], Matthieu Cord[a]

[a]*LIP6 UMPC, 104 avenue du Président Kennedy, 75016 Paris, France.*
[b]*L3I Université de La Rochelle, Avenue M. Crépeau, 17042 La Rochelle, France.*

Abstract

In this article, we introduce an application of swarm intelligence to distributed visual information retrieval distributed over networks. Based on the relevance feedback scheme, we use ant-like agents to crawl the network and to retrieve relevant images. Agents movements are influenced by markers stored on the hosts. These markers are reinforced to match the distribution of relevant images over the network. We tackle the use of the information gathered during previous search sessions. In order to match the different categories available on the network, we use several markers. Sessions searching for the same category will thus use the same makers. The system involves three learning problems: the selection of relevant markers regarding the searched category, the reinforcement of these markers and the learning of the relevance function. All of these problems are based on the relevance feedback loop. We test our system on a custom network hosting images taken from the well known *TrecVid* dataset. Our system shows a high improvement over classical content based image retrieval systems which do not use previous sessions information.

Key words: Distributed information retrieval, Content based image retrieval, Multi-agents system, Ant colony optimization

[*]Corresponding author, *tel* +33 1 44 27 88 18

Email addresses: `david.picard@lip6.fr` (David Picard),
`arnaud.revel@univ-lr.fr` (Arnaud Revel), `matthieu.cord@lip6.fr` (Matthieu Cord)

1. Introduction

The exponential growth of multimedia contents (*i.e.* images, videos or music) available on the Internet raises the question of the adaptability of searching tools to this highly distributed context. This is even more crucial when considering p2p networks: no centralized structure can gather all the information in the manner of classical web search engine. However, the decentralized nature of the production of multimedia content can have numerous advantages, such as the possibility of sharing computational and storage resources, the absence of data replication (and the questions of updating), etc. To benefit from such distributed structures, searching tools must be able to find the good location of relevant documents, *e.g.* to route the query among the peers correctly. This question has been widely studied in the p2p literature [20].

The question of image retrieval in this distributed context has been studied recently [5, 17]. Classically, in Content Based Image Retrieval (*CBIR*), visual features are extracted from the images and then compiled into an index or signature [22]. To perform the retrieval, a similarity function is computed to compare the index of the query to those of the collection. A ranking of the results is produced according to the similarity and shown to the users. To further improve the quality of the retrieval, an interaction with the user, called *relevance feedback* [23, 13], can be added. The user labels a fraction of the results, and the similarity function is updated consequently. Some machine learning techniques have been widely used in this case, and have shown very good improvements. Distributed *CBIR* is more challenging. In order to be efficient, distributed systems shall include the relevance feedback scheme but also a network exploration strategy [5]. Given a network of computers hosting images, systems must find the locations of relevant images, before any similarity learning. We have previously proposed such a *CBIR* extension to distributed sources based on biologically inspired multi-agents systems. In this system, the ant-like behavior of the agents leads to the optimization of the source selection in which we perform the retrieval, with successful results [16].

The main purpose of this paper is to deal with long-term memory of network based system for information retrieval. Indeed, the main drawback of classical Information Retrieval (*IR*) systems, is their absence of memory regarding previous queries. In the case of a widely used and highly distributed system, we dispose of numerous search sessions. The challenge is to be able

to re-use the knowledge brought by users in past search sessions to improve future sessions. In this context, the improvement can be done in two different ways: either by updating the similarity function according to the previous sessions, or by improving the selection of the images collections regarding the locations previously learned. In this paper, we undertake to improve the source selection stage by using the relevance feedback of past sessions only. This leads to a threefold learning problem : learning paths during the search session, merging paths learnt during previous search sessions, and learning the similarity function. We introduce a multiple-layered architecture to handle the three learning problems in a unified framework. We use an ant-like multi-agents system that learns the paths leading to relevant images collections regarding the searched category.

In the next section, we introduce a biologically inspired multi-agents systems (MAS) for information retrieval. In section 3, we describe how the system works for a single session, which we call our *"intra-session"* learning strategy. The use of several previous sessions to further improve the system, which is called the *"inter-sessions"* learning strategy, is shown in section 4. We then present extensive experiments done on an *ad hoc* network, using the *TrecVid'05* image dataset.

2. Multi-agents Systems for information retrieval

Distributed information retrieval (IR) systems need to perform three tasks [3], namely:

1. Selecting the collections in which performing the search.
2. Processing the documents on the selected collections.
3. Merging the results from the selected collections.

The second step is well known, as it does not differ from classical IR. The third step might contain difficulties, especially when the measure used to produce the results on each source are not directly comparable. This is often the case in distributed $CBIR$ where different collections may not use the same indexing techniques [1]. However, the first step is, to our minds, the more challenging. The ever growing number of sources forbids the exhaustive propagation of the query to each collection: Relevant sources must be identified so as to optimize the use of available resources. Typical problems involve optimal routing to minimize bandwidth waste and correct distribution of the processing tasks among the computational resources.

Multi-agents system have proved to be highly efficient for these tasks specific to distributed systems. Multi-agents systems are often used to parallel processes or to add robustness (by adding redundancy). In our work, we focus on *Mobile Agents*, a particular subset of *MAS* with nice properties for information retrieval. Mobile agents are autonomous computing software with the ability to migrate its executable code from one peer of the network to another and to continue its execution there [14]. Mobiles agents have been the focus of many researches in the late 1990s. There are good reasons to use mobile agents for information retrieval [15], such as the absence of a global structure in the network, the natural discovery of new collections, the automatic paralleling of the processing, etc. Recently, search engines based on mobile agents have been built [19, 18, 2, 11].

To further improve these agent based search engine, the next step is the introduction of swarm intelligence algorithms. *Ant Colony Optimization*[9] is an optimization strategy inspired by the behavior of ants while foraging. By modifying the environment, agents contribute to a collaborative solving of the problem [8]. It has been shown to be efficient in various optimization problems like *Quadratic Assignment Problem*, *Traveling Salesman Problem* [10], with applications to network routing, for instance [4]. In the case of information retrieval, many agents try to collectively find relevant documents during a search session leading to a better routing of the query, like in [18]. As there are many users of the network, the interaction of different sets of session related agents shall lead to further improvement.

In the specific case of distributed *CBIR* based on relevance feedback scheme, the three steps defined at the beginning of this section have to be done at each round. We expect the use of *ACO* like algorithm to be able to boost the source selection step by merging the information gathered by many agents (owned by several users) regarding the relevant hosts.

3. Intra-session learning strategy

In the hereby section, we recall the first system addressing distributed *CBIR* we developed [16]. The design is based on the relevance feedback scheme of classical *CBIR* tools, with the use of ant-like agents to handle the step 1 of distributed retrieval.

3.1. Selection strategy

To address the problem of the selection of sources, we introduce the *relevance r_i* of a host i, which is the number of relevant images it contains over the number of images in its collection. Thus, a collection with only relevant images has a relevance of 1, whereas a collection with no relevant image has a relevance of 0. The strategy we proposed consists in selecting a higher number of images from hosts with a high relevance. However, the relevance of each host cannot be known prior to the query, as we do not have the information of which image is relevant or not. Thus, we proposed to learn the relevance of the host in interaction with the user, using the relevance feedback loop. Let us define the *approximate relevance $\hat{r}_i$* of host i as the number of images from i which were labeled relevant by the user over the number images from i that were labeled by the user (regardless their labels). In fact, as the number of labels increases, the approximate relevance tends to the real relevance. During the relevance feedback loop, we select images from the hosts with a high approximate relevance, thus returning a higher number of relevant images.

The global strategy within a search session can be sum up as follows:

1. Let $\mathcal{A}(n)$ be the training set of images $\{\mathbf{x}_j\}_n$ and their respective labels $\{y_j\}_n$ at iteration n of the relevance feedback loop.
2. According to $\mathcal{A}(n)$ the approximate relevance $\hat{r}_i$ and the relevance function $f_{\mathcal{A}(n)}$ are computed.
3. Using $f_{\mathcal{A}(n)}$ and $\hat{r}_i$, I images $\{\mathbf{x}_j\}_{n+1}$ are selected, with a number of images proportional to $\hat{r}_i$ selected from i, $\forall i$.
4. The training set for the next iteration is built as $\mathcal{A}(n+1) = \mathcal{A}(n) \cup \{\mathbf{x}_j, y_j\}_{n+1}$, where y_j are the labels given by the user to the I selected images $\{\mathbf{x}_j\}_{n+1}$.

3.2. System overview

Figure 1 describes our system. The user begins the query by giving an example or a set of examples to an interface (1). A similarity function based on these examples is built (2). To build the similarity, visual features are extracted from the examples (distributions of colors and textures obtained by quantization of color and texture spaces over a set of training images), and machine learning techniques such as *SVM* are used [7, 21]. Mobile agents

5

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. These agents travel through the network according to movement rules related to numerical markers present on each host reflecting the relevance of the hosts (see below 3.3) (4).
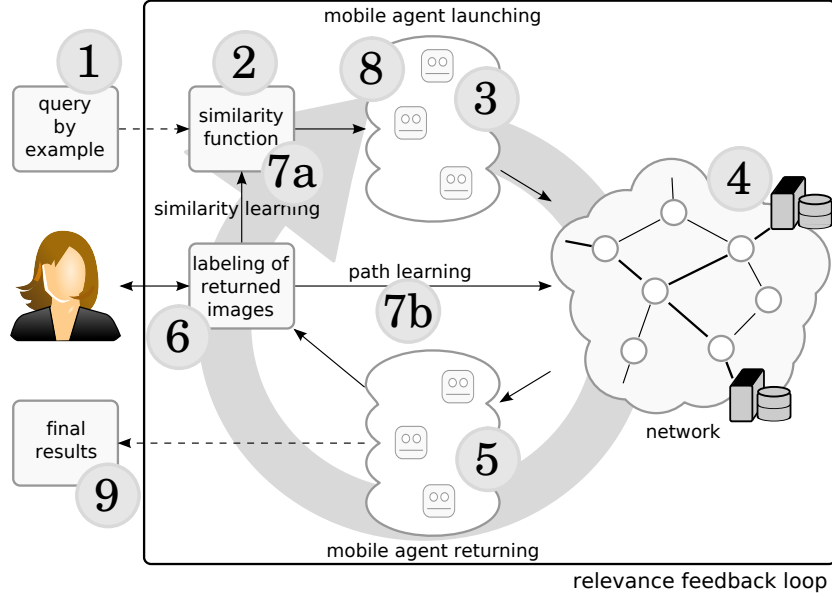

Figure 1: Functional description of our system showing the user in interaction with the relevance feedback loop (launching of agents, retrieval, display and labeling).

On each platform, an agent indexing the local images is run. The incoming mobile agent sends the similarity function to the index agent which returns the most informative images.

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 [7], the update stage 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. The interactive loop consists in several launching of mobile agents and labeling of the results (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. In combination with the ant-algorithm, this assures that most of the best retrieved images are provided by relevant hosts.

3.3. Learning the relevance of the hosts

The main key point of the introduced strategy relies on the estimation of the relevance of each host of the network at each relevance feedback round. As the number of hosts increases, it is not possible to visit them exhaustively to compute the approximate relevance anymore. Instead, we propose a stochastic exploration of the network, where relevant hosts are prioritized using and adaptation of ACO. Each host of the network is provided a marker (a numerical value) which acts as the pheromone used in ant-like strategies. Let us denote m_i the marker associated with host i. Then, the probability for an agent to move from host s to host i is

$$P_{s \to i} = \frac{m_i}{\sum_{k \in dest} m_k} \qquad (1)$$

Where $dest$ is the set of possible destinations from s. Thus, the higher the value of the marker, the higher the probability for an agent to move there.

Markers are reinforced each time an agent returns to reflect the relevance of hosts, using the following rules:

$$m_i \leftarrow \alpha m_i + \beta a_i + \gamma u_i \qquad (2)$$

Where $a_i = 1$ if i contains an image collection (0 if not), and $u_i = 1$ if the image the agent has returned is labeled relevant (0 if not). $0 \leq \alpha < 1$, $\beta \geq 0$ and $\gamma \geq 0$ are parameters of the system. With these reinforcements of markers, the probability for an agent to move to a host is directly linked with its approximate relevance (see section 5).

4. Inter-sessions learning strategy

As stated in the introduction, classical $CBIR$ tools lack the reuse of the information injected in the system by the user through the relevance feedback loop. For instance, suppose several users have searched for images of *cars*, *mountains* or *sporting events*, each session totalizing 100 pairs of images and associated labels (in the form of either *relevant* or *irrelevant*). If we assume that different users have a consistent definition of the categories they

are searching (*i.e.* they consider the same images as being relevant when searching for the same category), then the use of these previous labels can be a precious boost to the learning strategies involved in the retrieval of these three categories for later sessions.

Yet the problem is difficult. Since we have no other indication on the subject of a session than the set of labels gathered during the feedback loop, we have no *a priori* knowledge on which previous labels could be used for a new session. This weakly supervised learning problem has not been widely studied. The main development in this direction have been done regarding the update of the similarity between images to match our perception of similarity better. In the case of kernel based similarity, it can be done by learning the similarity matrix [12, 6]. To our knowledge, there has been no work on long-term distributed *CBIR*.

Thus, we propose to improve the source selection stage using long-term learning: within a search session, paths leading to relevant images are learned using relevance feedback. We intend to reuse previously learned paths for later sessions, so as to speed up the time taken to interactively learn the location of relevant images.

As there might be several categories of images the users can be looking for, a single marker on each host is not sufficient to match the relevance of hosts for all searched category. We propose to store P markers on each host, each one being related to a category. This forms a vector of dimension P, called $\mathbf{m}_i$ for host i. We call the set of all markers $\{\mathbf{m}_i[c], i \in N\}$ of the network N associated with a category c a *plane*. Let us denote p_c the plane related to concept c, and P the total number of planes. The goal of long term learning is then to find values for each vector, such that each component of the vector reflects the relevance of the host regarding the corresponding category (as in figure 2).

With such an architecture, the retrieval involves three learning problems:

1. Selecting the plane corresponding to the searched category.
2. For the selected plane, improving the marker values so as to match the relevance of hosts.
3. Learning the relevance function.

The first and the third steps have to be done during a search session, whereas the second step can be done over many sessions by many users.
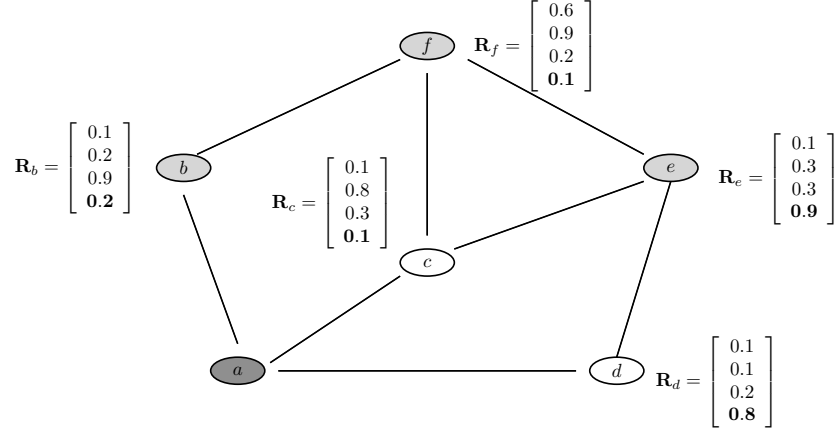
Figure 2: Example of network with 6 hosts: the user's computer (a), 3 hosts with an image collection (b, e and f) and two routing hosts (c and d). The system has $P = 4$ planes available. For instance, agents using the fourth plane (bold values) have more chance to be routed to host d and to retrieve images from host e.

4.1. Learning strategies

We focus on the first two learning problems. The third one is exactly the same as for the previously presented system, and is not detailed here. The selection of a plane and the learning of markers problems are interdependent. In fact, it is only possible to select a relevant plane if the markers have been learned and match the relevance of hosts regarding the considered categories. But is is also only possible to learn the markers if a plane has been selected. To tackle this dependence, we propose an iterative system that improves both stages at every step of the relevance feedback.

Let us first consider the plane selection, and suppose that markers have been correctly learned over past sessions. We define a selection function ψ_s that given a marker vector $\mathbf{m}_i$ returns the component p corresponding to category s:

$$
\begin{aligned}
\psi_s : \quad & \mathbb{R}^{P'} \to \mathbb{R} \\
& \mathbf{m}_i \mapsto \mathbf{m}_i[p]
\end{aligned}
\tag{3}
$$

To produce a function ψ_s, we assign a weight $w_j, 0 \leq j \leq P$ to each component of the markers vector space. The component p is sampled using the multinomial law $\mathcal{M}(1; w_1, ..., w_P)$. As we have no *a priori* about which

component is relevant to the searched category, we initialized the weights to $1/P$. We then propose to learn the weights w_j thanks to the relevance feedback loop. Each time the user labels an image retrieved using the plane p, we update w_p as follows:

$$w_p \leftarrow w_p + \varepsilon(u - w_p) \tag{4}$$

Where u is as defined in 3.3, and ε is the learning rate. A new selection function is sampled each time an agent is launched in order to explore different planes before focusing on the most relevant. The learning rate is set to ensure convergence of the selection within a few iterations of relevance feedback.

Let us now consider the learning of markers, and suppose the relevant plane has been selected. As we have only one value per host, we have an optimization problem similar to the one presented in section 3 (see equation 2). Thus, we can use the same reinforcement rules so as to match the values of the selected plane with the relevance of the corresponding hosts regarding the searched category. The only difference with the previous system is that the markers are now updated over several sessions (issued by different users), improving the approximate relevance of the hosts.

4.2. Relevance feedback detail

The detail of an iteration of the relevance feedback loop can be described as follows:

1. Let $\mathcal{A}(n)$ be the training set of images $\{\mathbf{x}_j\}_n$ and their respective labels $\{y_j\}_n$ at iteration n.
2. A selection function $\psi_{\mathcal{A}(n)}$ is generated with the help of $\mathcal{A}(n)$, that selects the plan p.
3. Thanks to p and $\mathcal{A}(n)$, the approximate relevance $\hat{r}_i$ is computed based on the markers $\mathbf{m}_i[p]$ of plane p.
4. A relevance function $f_{\mathcal{A}(n)}$ is trained upon $\mathcal{A}(n)$.
5. $f_{\mathcal{A}(n)}$ and the approximate relevance $\hat{r}_i$ are used to select I images $\{\mathbf{x}_j\}_{n+1}$, such that the number of images retrieved from i is proportional to $\hat{r}_i$.
6. The next training set is defined by $\mathcal{A}(n+1) = \mathcal{A}(n) \cup \{\mathbf{x}_j, y_j\}_{n+1}$, where y_j are the labels given by the user to the retrieved images $\mathbf{x}_j$.

At each step of the relevance feedback, all the three problems (namely the selection of plane, the reinforcement of markers and the learning of the relevance function) are improved. The long-term improvement comes from the reinforcement of the markers $\mathbf{m}_i[p]$ over several session, leading to a better approximation of the relevance of hosts. This improves the source selection stage of the distributed retrieval problem.

5. Theoretical Aspects

The three learning problems involved in our system are entangled, since they are needed to perform a single iteration of the relevance feedback loop. Moreover, the searched categories tend to a competitive specialization of the available planes due to the lack of prior knowledge on the location of categories. Thus, the overall dynamics are very complex. However, we can theoretically study the simpler case of a single plane used for a single category. In that case, we can show that the system converges to the reinforcement of paths leading to the relevant images regarding the category, with a close relation to the approximate relevance of hosts. We derive the following theorem:

Theorem 1. *Let $m_i(n)$ be the marker of host i at time n, then, it converges in probability to $E[m_i]$:*

$$\forall \varepsilon, \forall \delta, \exists N \mid \forall n \geq N \Rightarrow P(|m_i(n) - E[m_i(n)]| > \varepsilon) < \delta \qquad (5)$$

To prove the theorem, we start by using Markov inequality:

$$\forall \Delta > 0, P(Y > \Delta) \leq \frac{E[Y]}{\Delta} \qquad (6)$$

on variable err defined by :

$$err = |m_i(n) - E[m_i(n)]| \qquad (7)$$

then

$$\forall \varepsilon, P(err > \varepsilon) \leq \frac{E[err]}{\varepsilon} \qquad (8)$$

$$\leq \frac{E[m_i(n) - E[m_i(n)]]}{\varepsilon} \qquad (9)$$

$$\leq \frac{E[\alpha^{n-1} m_i(0) + \sum_0^{n-1} \alpha^{n-1-k} \Big(\beta a_i(k) + \gamma u_i(k)\Big) - E[m_i(n)]]}{\varepsilon} \qquad (10)$$

11

$$\leq \frac{\alpha^{n-1}m_i(0) + \sum_0^{n-1}\alpha^{n-1-k}\left(\beta E[a_i] + \gamma E[u_i]\right) - E[m_i(n)]}{\varepsilon} \qquad (11)$$

We remark that

$$\forall n, \sum_0^{n-1}\alpha^{n-1-k} < \frac{1}{1-\alpha} \qquad (12)$$

thus

$$P(err > \varepsilon) < \frac{\alpha^{n-1}m_i(0) + \frac{\beta E[a_i] + \gamma E[u_i]}{1-\alpha} - E[m_i(n)]}{\varepsilon} \qquad (13)$$

From equation 2, we can express $E[m_i(n)]$ as:

$$E[m_i(n)] = \alpha^{n-1}m_i(0) + \beta E[a_i]\sum_0^{n-1}\alpha^{n-1-k} + \gamma E[u_i]\sum_0^{n-1}\alpha^{n-1-k} \qquad (14)$$

$$= \alpha^{n-1}m_i(0) + \beta E[a_i]\sum_0^{n-1}\alpha^{k} + \gamma E[u_i]\sum_0^{n-1}\alpha^{k} \qquad (15)$$

$$= \alpha^{n-1}m_i(0) + (\frac{1-\alpha^{n-2}}{1-\alpha})(\beta E[a_i] + \gamma E[u_i]) \qquad (16)$$

Replacing the expression of $E[m_i(n)]$, we obtain:

$$P(err > \varepsilon) < \alpha^{n-2}\frac{\beta E[a_i] + \gamma E[u_i]}{\varepsilon(1-\alpha)} \qquad (17)$$

then, $\forall \delta$, one can always choose N such that:

$$\alpha^{N-2} < \frac{\varepsilon(1-\alpha)}{\beta E[a_i] + \gamma E[u_i]}\delta \qquad (18)$$

it follows that $\forall n > N$:

$$P(err > \varepsilon) < \alpha^{N-2}\frac{\beta E[a_i] + \gamma E[u_i]}{\varepsilon(1-\alpha)} \qquad (19)$$

$$P(err > \varepsilon) < \delta \qquad (20)$$

Thus we have shown that:

$$\boxed{\lim_{n\to\infty} P\big(m_i(n) = \frac{\beta E[a_i] + \gamma E[u_i]}{1-\alpha}\big) = 1} \tag{21}$$

We can also express the approximate relevance of host i using u_i. In fact, as the approximate relevance is the ratio of positive labels over the total number of label from i, it is directly equal to $E[u_i]$. Thus, using equation 21 for the special case $\beta = 0$, the probability of movement after a sufficient evolution of all markers is:

$$P_{s\to i} = \frac{m_i}{\sum_{k\in dest} m_k} \tag{22}$$

$$\approx \frac{\gamma E[u_i]}{1-\alpha} \frac{1-\alpha}{\sum_{k\in dest} E[u_k]} \tag{23}$$

$$\approx \frac{E[u_i]}{\sum_{k\in dest} E[u_k]} \tag{24}$$

Which is a normalized value of the relevance of host i. As many agents are launched (with the goal of retrieving a fixed number of images), the total number of images retrieved has a distribution over the hosts proportional to their relevance. Hence, the higher probability of retrieving images is from the most relevant hosts.

6. Experiments

In the hereby section, we present experiments made on custom networks of computers hosting images taken from the *TrecVid'05*[1] keyframes dataset. These networks have a low connectivity, which is also often the case of real world peer-to-peer networks. These experiments aim at testing the inter-session system. A comparative experiment of the intra-session system (which measures the gain of using the location of the relevant data) can be found in [16]. We first introduce the setup used for the experiments, and then evaluate the system's overall performances. We further investigate the dynamics of our system.

---

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

6.1. Experimental setup

The dataset used consists in about 70000 images taken from the TrecVid'05 challenge. Images are extracted from videos of TV shows and are classified as various categories such as objects (bus, airplanes, ...), scenes (mountains, urban, ...), or more abstract concepts (explosion and fire, entertainment, charts, ...). The features we use consist in vectors containing 32 bins of colors and 32 bins of textures as described in [16].

We set two different configurations for our experiments, with different networks and category distribution. The first setup, noted A, tests the performances of the system. The second setup, noted B, investigates the evolution of the markers.

Setup A

Setup A is made of five hosts. The first computer is the starting point of the query, where agents are launched and retrieved. The four other computers are hosting image collections. Their contents are chosen so as to be visually similar. Only 10 categories are used for this experiment.

The parameters of this setup are as follows :

- For each retrieval session, feedback loops are conducted until 100 labels are gathered.

- At each new retrieval session, the search category is randomly selected. In total, 70 search sessions are made for each category.

- At each feedback loop, 8 agents are launched, with the goal of retrieving 2 images.

- 8 planes are used.

There are 10 categories in this setup, but as there are only four hosts, we assume that 8 planes are sufficient to match the distribution of the categories over the network. Hence, a plane might well be associated with two different categories having exactly the same distribution on the network.

Setup B

Setup B consists in only three hosts containing an image collection, plus the computer initiating the queries. Three categories, namely *airplanes, maps*

and *explosion-fire*, are used as search categories and each is hosted on a specific computer. These categories do not share any image, hence no confusion about the location of the category can be made while retrieving a positive label. A fourth set contains 4000 images from the category *entertainment*, which has major visual variations in its content, and thus could be visually associated with any of the other categories. It is used as additive noise to the specialization of the categories to reflect the difficulties to find the relevant images in a collection containing varying topics. It is uniformly distributed among the three other hosts.

For this setup, parameters are the following :

- For each retrieval session, feedback loops are conducted until 100 labels were gathered.

- At each new retrieval session, the search category is randomly selected. In total, 70 search sessions are made for each category.

- At each feedback loop, 8 agents are launched, with the goal of retrieving 2 images.

- 6 planes are used.

As there are only three hosts, each one hosting a single searchable category, we assume 6 planes are sufficient to match the distribution of the categories.

6.2. Overall performances

In this test, we evaluate the overall performances of the system. For this purpose, we use the setup *A*. At the end of each session, we measure the recall on the images retrieved (number of relevant images retrieved over the number of available relevant images). Results are shown on figure 3, and compared to results obtained without using long-term learning, as described in [16].

We can see a significant improvement as high as 100% of the recall due to the better learning of the localizations of the categories. The complexity of the category (which can be evaluated by the absolute value of the recall) has no effect over the improvement of the long-term learning.
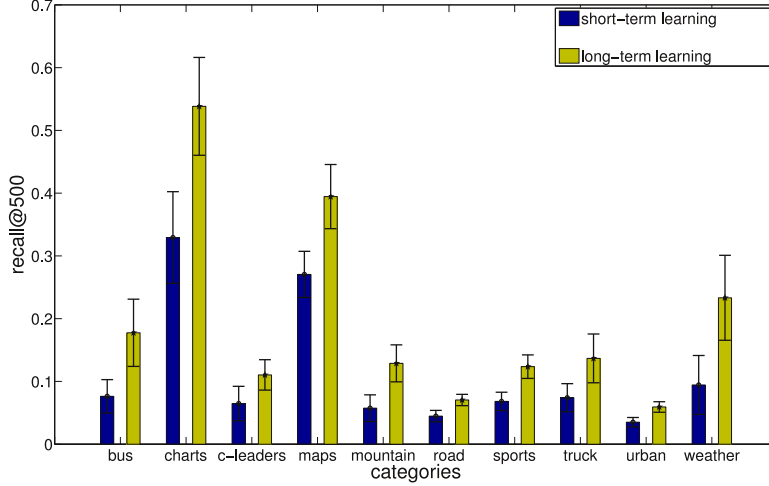
Figure 3: Recall obtained for the experiments of setup A. The long-term improves the recall for all categories by a significant margin.

6.3. Markers evolution

In order to analyze the system better, we measure the evolution of the markers during the search sessions using setup B. We first compute the average probability over all search sessions of each plane to lead to one of the three destinations. Results are shown on figure 4. As one can see, the planes are specialized to one destination. For instance, the first plane has a higher probability to lead to host one, while the second plane has a higher probability to lead to host 3. As the values of these probabilities are clearly above 0.5, we consider the learning of the localization as successful.

We then observe the evolution over time of the probabilities of movement by plotting them at the end of each session. On figure 5, we introduce a visualization where each mark shows the probability of movement of the corresponding plane, projected on three axes given the following equation:

$$(x_j, y_j) = (\sum_{i=0}^{2} \mathbf{m}_i[j] \cdot \cos(\frac{i \cdot 2\pi}{3}), \sum_{i=0}^{2} \mathbf{m}_i[j] \cdot \sin(\frac{i \cdot 2\pi}{3})) \qquad (25)$$

With (x_j, y_j) being the coordinate of the considered mark.

As seen on figure 5, some planes are specialized during all sessions (for instance plane 1), while some migrate their specialization from one host to
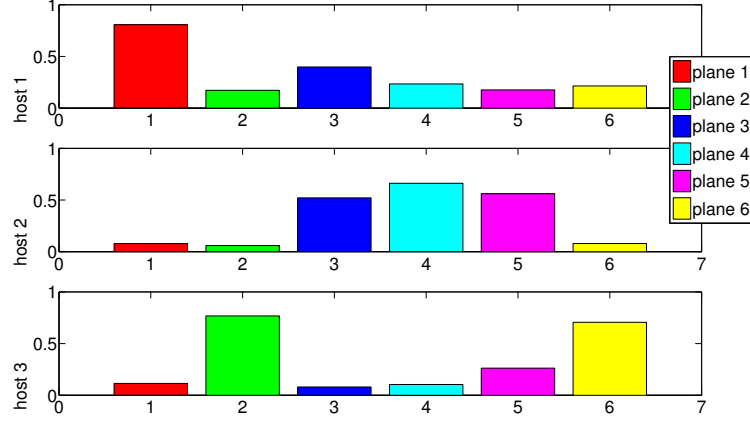
16

Figure 4: Average of the probability of leading to each host for each plane in setup B. For instance, the plane 2 has a higher probability to lead to host 3.
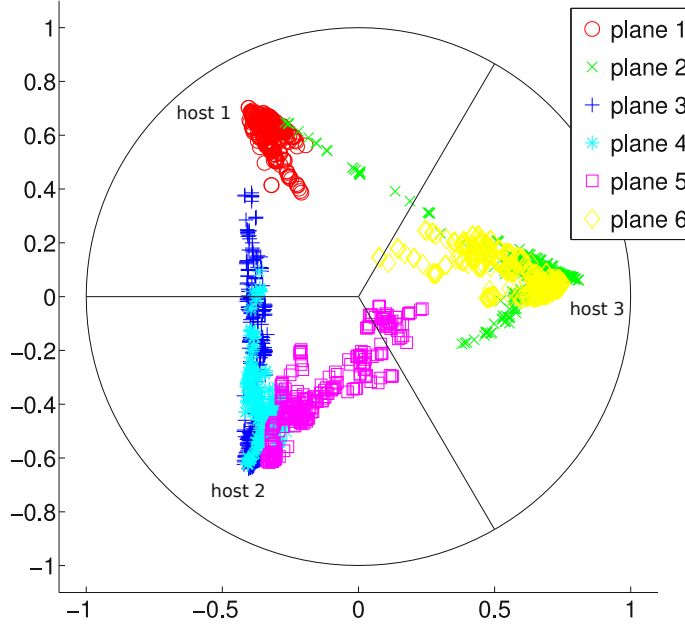

Figure 5: Distribution of the markers on the three hosts at the end of each session in setup B. Each plane has specialized towards a specific destination for the major part of the experiment.

17

another (for instance, plane 3), explaining their lower average value on figure 4.

We then investigate the dynamics of the markers evolution. for that purpose, we plot the value of a marker on a particular host at an instant $t + 1$ against its value at time t. Such plots are shown on figure 6 for the three hosts.

We can see that the evolution are close to the diagonal, which means that changes are smooth. There are oscillations around a fixed point (for instance $(0.8, 0.8)$ for the first plane on the first host), proving experimental stability of the association between planes and markers. Some planes (namely 3 and 5) take a longer time to reach a stable association, but end oscillating around a high probability value on one host.

7. Conclusion

In this article, we introduced an application of swarm intelligence to distributed content based image retrieval. The hereby presented system relies on mobile agents with ant-like behavior for optimizing their route through the network in search of relevant images. A relevance feedback loop is used to improve both the relevance function and the paths to relevant image collections. We tackle the problem of using previous search sessions to improve future sessions by adding several markers on the hosts, each one being associated with a category. This multiple markers strategy raises a three layered learning problem, namely:

1. Selecting the relevant set of markers regarding the searched category.
2. Optimizing the selected markers so as to match the distribution of relevant images.
3. Learning the relevance function.

As these problems are interdependent, the resulting system has complex dynamics. We introduced new learning rules dealing with this complexity. Theoretical results on the convergence of these rules have been derived for a simple case.

We also carried out experiments on custom networks hosting images taken from the *TrecVid'05* dataset. The long-term learning strategy results in a high improvement over classical distributed *CBIR*. We investigated the evolution of the markers, and showed that the association of a plane with

18
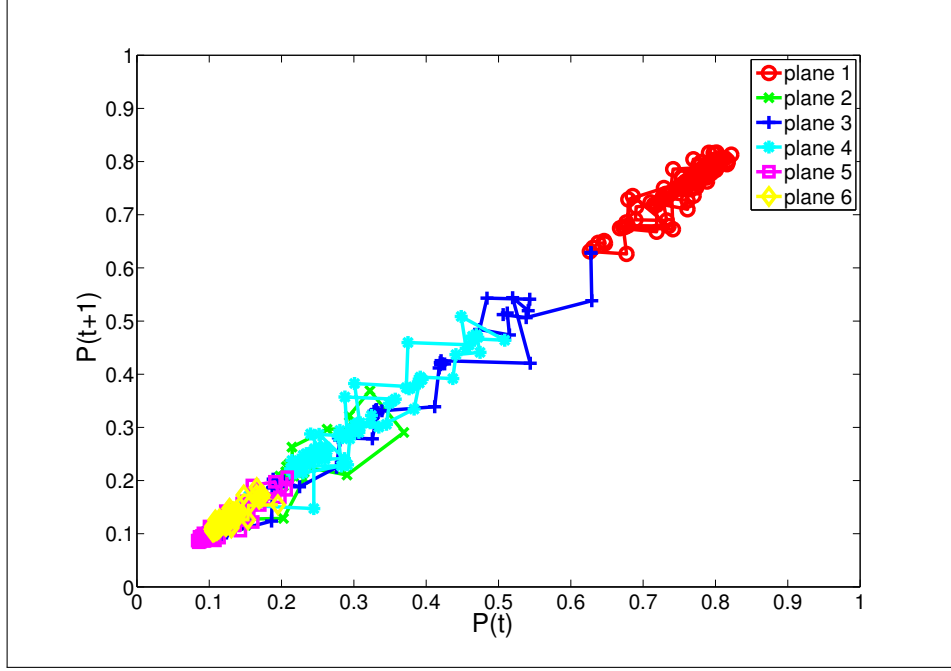
(a) Evolution of the markers between each sessions for host 1. Marker of the first plane converges to $(0.8, 0.8)$ and oscillates around this point, whereas other markers converge and oscillate around $(0.1, 0.1)$.
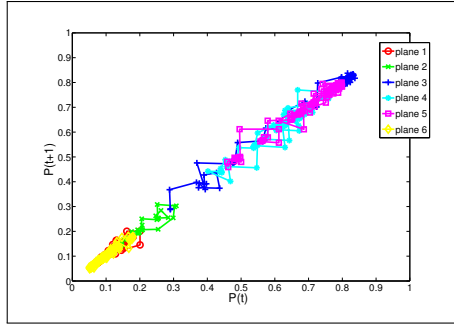
(b) Evolution of the markers between each sessions for host 2. Markers of the planes 3, 4 and 5 converge slowly to $(0.8, 0.8)$.
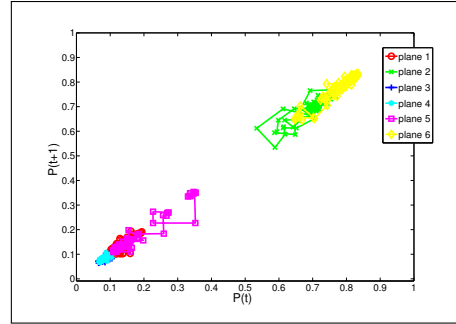
(c) Evolution of the markers between each sessions for host 3. Markers of planes 2 and 6 converge towards $(0.8, 0.8)$.

Figure 6: Evolution of of markers of the six planes for the three hosts. For each of these hosts, one or more plane converges to $(0.8, 0.8)$ and oscillate around this point.

a category was correctly learned. Furthermore, we found these associations are stable and the transitions are smooth.

Perspectives on this work include testing on web-scale networks. The challenges of these large networks is on one hand the scaling of the reinforcement based learning algorithm (despite ACO has been shown to scale well). On the other hand, it is necessary to keep the system performing in real-time, which is needed for the relevance feedback scheme.

References

[1] Stefano Berretti, Alberto Del Bimbo, and Pietro Pala. Merging results for distributed content based image retrieval. *Multimedia Tools Appl.*, 24(3):215–232, 2004.

[2] F.M.T. Brazier, D.R.A. de Groot, A. Oskamp, and N.J.E. Wijngaards. Agent-based information retrieval: Legal and technical considerations in a simple case. pages 95–107, June 2004.

[3] J. Callan. Distributed information retrieval. *Advances in information retrieval*, 5:127–150, 2000.

[4] G. Di Caro and M. Dorigo. AntNet: a mobile agents approach to adaptive routing. Technical Report IRIDIA/97-12, Université Libre de Bruxelles, Belgium, dec 1997.

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

[6] 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.

[7] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[8] J.L. Deneubourg and S. Goss. Collective patterns and decision-making. *Ethology Ecology & Evolution*, 1:295–311, 1989.

[9] M. Dorigo, M. Birattari, and T. Stützle. Ant colony optimization artificial ants as a computational intelligence technique. *IEEE Comput. Intell. Mag.*, 1(4):28–39, 2006.

[10] Marco Dorigo and Luca M. Gambardella. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66, April 1997.

[11] D.R.A. de Groot, M.L. Boonk, F.M.T. Brazier, and A. Oskamp. Issues in a mobile agent-based multimedia retrieval scenario. In *Proceedings of The 4th Workshop on the Law and Electronic Agents (LEA 2005)*, pages 33–43, June 2005.

[12] D. R. Heisterkamp. Building a latent semantic index of an image database from patterns of relevance feedback. In *International Conference on Pattern Recognition (ICPR)*, Quebec City, Canada, 2002.

[13] 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)*, volume 3, pages 2–5, Thessaloniki, Greece, October 2001.

[14] Danny B. Lange. Mobile objects and mobile agents: The future of distributed computing? In *ECOOP*, pages 1–12, 1998.

[15] Danny B. Lange and Mitsuru Oshima. Seven good reasons for mobile agents. *Commun. ACM*, 42(3):88–89, 1999.

[16] D. Picard, M. Cord, and A. Revel. Image retrieval over networks : Active learning using ant algorithm. *IEEE Transactions on Multimedia*, 10(7):1356–1365, nov 2008.

[17] 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.

[18] 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.*, pages 482–485, Halifax, Canada, October 2003. IEEE, IEEE Computer Society.

[19] Volker Roth, Ulrich Pinsdorf, and Jan Peters. A distributed content-based search engine based on mobile code. In *SAC '05: Proceedings of the 2005 ACM symposium on Applied computing*, pages 66–73, New York, NY, USA, 2005. ACM Press.

[20] Kunwadee Sripanidkulchai, Bruce Maggs, and Hui Zhang. Efficient content location using interest-based locality in peer-to-peer systems. In *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (Infocom 2003)*, San Francisco, March 2003.

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

[22] R.C. Veltkamp. Content-based image retrieval system: A survey. Technical report, University of Utrecht, 2002.

[23] 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*, pages 13–20, Bristol, UK, September 1998.

David Picard received the Ph.D. in image processing from the University of Cergy-Pontoise, France, in 2008. He is currently pursuing a postdoc at the LIP6 Laboratory, UPMC, Paris, France. He is a Graduate Engineer in computer science at ENSEA, France. His research interests concern content based image retrieval in a distributed context using cooperative systems and bio-inspired algorithms, as well as machine learning techniques for multimedia retrieval.

Arnaud Revel is a Graduate Engineer of the Ecole Nationale Supérieure de l'Electronique et de ses Applications (ENSEA) de Cergy-Pontoise (graduated in computer science) and received the Ph.D. degree from the University of Cergy-Pontoise, France, in 1997 and his Qualification to Research Direction in 2007 from the same University. He has currently a full professor position at the La Rochelle University. His research interests are to develop neural architectures and learning algorithms inspired by biology and psychology in order to control autonomous agents (either robots, software or in virtual reality). He is particularly interested in modeling and implementing interaction capacities in robotic and virtual agents in order to develop intuitive a human machine interface and provide new therapeutic tools for psychiatry. He has also been developing multi-agent systems for information retrieval for several years.

Matthieu Cord obtained his Ph.D. degree in Image Processing in 1998 by the University of Cergy-Pontoise, France, and was a postdoc in 1999 at the Katholieke Universiteit Leuven, Belgium. Then, he joined the ETIS labs in France to create the image indexing research group. In 2006, he joined the UPMC-Paris 6 University, where he got a full professor position. He is involved in several French and international research programs and projects and has been recently nominated to the prestigious French Research Institute (IUF) for 5 years. His research interests include Computer Vision, Image Processing, Machine Learning and their applications to Multimedia Information Retrieval and Multimedia Processing.