A Multimodal query expansion based on genetic programming for visually-oriented e-commerce applications

Patrícia C. Saraiva[a,*], João M. B. Cavalcanti[a], Edleno S. de Moura[a], Marcos A. Gonçalves[b], Ricardo da S. Torres[c]

[a]*Institute of Computing, Federal University of Amazonas, AM, Brazil*
[b]*Department of Computer Science, Federal University of Minas Gerais, MG, Brazil*
[c]*Institute of Computing, University of Campinas, SP, Brazil*

Abstract

In this work, we present a multimodal query expansion strategy based on genetic programming for image search in visually-oriented e-commerce applications. The proposed method automatically expands image queries using multimodal information and computes a ranking of results based on the expanded query. Genetic programming is used to both learn to expand and learn to compute the ranking for the expanded queries. In order to evaluate the performance of our method, we have collected two datasets containing clothing images of products taken from different online shops. Experimental results indicate that our method is an effective alternative for improving quality of image search results. When compared to a genetic programming system based only on visual information, our method achieved gains of at least 19% (and up to 51%) in all collections and scenarios considered. Further, we compared the results of our method to a related system that, similar to ours, exploits additional types of evidence, but in a completely *ad-hoc* way, achieving gains of up to 54% in Mean Average Precision.

Keywords:
CBIR, Multimodal Query Expansion, Genetic Programming

*Principal corresponding author
Email addresses: patricia.saraiva@gmail.com (Patrícia Correia Saraiva), john@icomp.ufam.edu.br (João Marcos B. Cavalcanti), edleno@icomp.ufam.edu.br (Edleno Silva de Moura), mgoncalv@dcc.ufmg.br (Marcos Gonçalves), rtorres@ic.unicamp.br (Ricardo da S. Torres)

1. Introduction

Recent technological advances have contributed to create new opportunities for content-based image retrieval (CBIR) applications. This is the case of systems for searching products in online stores, popularized by the large usage of mobile devices, such as tablets and smartphones that can access applications using remote resources, as well as by modern and secure payment mechanisms that allow users to buy goods using their mobile devices.

Contributing to this scenario, we can see an increasing number of e-commerce companies which demand constant improvement in search technologies to support their operations. Another evidence that this specific problem is increasing in importance can be seen in recent reports summarizing e-commerce activities. These reports show a fast growing market share of segments such as fashion, apparel, and accessories along with furniture and home furnishings, where visual search can play an important role. For instance, these two last segments represent together about 25% of the e-commerce sales in the US in 2012, according to the eMarketer report (eMarketer, 2012). Garment, including clothes and accessories, is the second biggest category in the e-commerce shopping in the US and is growing in importance in most of the largest e-commerce markets world wide.

In this context, we propose a new machine-learning CBIR approach for a visually-oriented e-commerce applications where image search is one of the key components. In this specific domain, the use of an image for searching a particular item is much more descriptive than a textual query. A common scenario could be, for instance, when the user takes a photo of a product in a store with his cell phone and wants to find similar items to the one he found at the store. This type of query is useful when the user is searching for products such as clothing, shoes, handbags, watches, and accessories. The visual presentation of this sort of products is essential to the consumer purchase decision. Considering that usually there are several products visually similar in the product database, visual search can help in the specific task of finding the desired product and improving the user experience.

One of the key differences that make our target application different from the traditional image search applications is the final goal for the search task. While we have the query provided as an image, it still represents a search for products, which have other attributes beside their images. Thus, sometimes other products may be similar to the query, when considering the user interest, even though their image representation is not so close to the image provided in the query. Real-world examples that may cause the image of the product different from the query, even

though the product is relevant to the query include differences in the approach to produce the product photos, such as folded versus unfolded t-shirts. Products with distinct colors and textures may also be considered relevant in specific cases, given their styles and so on. We here provide a solution that allows finding relevant products related to an image query even though their image representation is not similar to the query. Our solution to the problem is to perform a multimodal expansion, using the initial query to infer other attributes that are relevant to the query, such as category and the textual description of the products. The key idea is to use the visual information to produce an initial ranking and then extract accurate multimodal information from the results that may be used as an expansion to the initial query.

Our strategy exploits a machine learning self-reranking approach based on automatic multimodal query expansion. The challenge here is that, different from most previous work, we do not assume that the query is defined in terms of both image visual content and textual descriptions. In fact, the constraints usually imposed by our target application imply that only the visual information is available, i.e. a photo of the desired product. The key idea of our method is to expand the initial image query with information about the inferred category of the query image along with textual content automatically associated with other similar images in terms of visual appearance. This type of information is usually largely available in online product catalogs, helping us in the goal of improving the quality of the overall retrieval process. Thus, we not only expand the initial query image with multimodal information, but also produce a new ranking based on the expanded query.

The expansion and actual ranking of images exploit a Genetic Programming (GP) approach to perform both the expansion of the initial query and the computation of a new ranking based on it. We propose and experiment four alternative ways of using GP for deriving multimodal query expansion methods. GP is used to find the best possible multimodal combination of the available pieces of evidence. We chose GP for a number of reasons, including: (i) excellent effectiveness in previous CBIR studies (Andrade et al., 2012; Calumby et al., 2012; Faria et al., 2010; Ferreira et al., 2011; Torres et al., 2009), mainly when exploiting multimodal information; (ii) capability of finding near-optimal solutions in large search spaces, as it is the case here; (iii) and capability of dealing with multiple objectives at the same time (in our case query expansion and effective ranking). After learning a ranking function in an offline process with GP, we apply the function at query processing time without any extra overhead. As far as we know, GP has never been applied in the scenario we deal in this article, in which only the visual information

is initially available for multimodal query expansion. The challenge of exploiting only visual aspects relies on the difficulty of mapping low-level features obtained by means of image processing algorithms to high-level concepts found in images, the well-known *semantic gap* problem (Liu et al., 2007).

Experimental results indicate that our new GP multimodal query expansion approach is able to significantly improve the overall quality of results of e-commerce visual search applications when compared to the application of GP without expansion. This demonstrates that the idea of performing a multimodal ML-based automatic expansion for image queries is very promising.

This work extends our previous solution (dos Santos et al., 2013), which, despite addressing a similar problem, i.e., searching products using only image queries[1], it does it in a completely ad-hoc way, not exploiting any type of learning paradigm. ML solutions are both: more principled, with a lot of theoretical background, as well as more flexible, being able to easily accommodate other types of multimodal evidence when available. Not only this, but our experimental results show that the GP-based solution outperforms our previous efforts in up to 54% when considering Mean Average Precision results.

This article is organized as follows. In next section, we describe related work regarding image search and product image search, including automatic expansion techniques previously proposed in the literature. Section 3 presents the visual features and the image datasets we used in our work. Section 4 presents an overview about the genetic programming approach and discusses the method we propose for expanding and re-rank image queries. Section 5 presents the experiments we have performed to assert the impact of the proposed method. Finally, Section 6 concludes the article while introducing possible future work directions.

2. Related Work

Image retrieval has been extensively studied in the last years. Kherfi et al. (2004) provide a comprehensive survey on Web image retrieval systems, giving details on the main issues that have to be addressed during their implementation (e.g., how to perform data gathering, visual feature extraction, indexing, retrieving, and performance evaluation).

Next we discuss research work on visual search which involves one or more of the following techniques: genetic programming, re-ranking, use of multimodal

[1] In fact, as far as we know, this is the only work that takes advantage of multimodal information to expand visual queries. Other works use multimodal information to expand textual queries

evidence and product visual search. At the end of this section, we discuss the work used as baselines.

2.1. Related Work using Genetic Programming

Torres et al. (2009) introduce the use of Genetic Programming (GP) for content-based image retrieval (CBIR). The proposed framework combines simple descriptors and exploits the GP principles to discover an effective combination function to retrieve images based on shape information. Experiments showed that the GP framework yields better results than a Genetic Algorithm (GA) approach, being able to find better similarity functions than the ones obtained by individual descriptors.

Three learning algorithms, CBIR-SVM, CBIR-GP, and CBIR-AR, are used in Faria et al. (2010) to effectively combine multiple image content descriptors using Support Vector Machines (SVM), Genetic Programming (GP), and Association Rules (AR), respectively. The objective is to improve ranking performance in content-based image retrieval tasks. The experiments showed that CBIR-GP and CBIR-AR have similar performance, and both outperformed CBIR-SVM generating a better image ranking function. In a different context, GP was used in (Andrade et al., 2012) to combine local and global descriptors aiming to support image and video retrieval tasks.

Piji and Jun (2009) present a ranking model named WIRank based on GP to automatically generate effective ranking functions by combining different types of evidence for web image retrieval. Their GP approach includes textual metadata, visual features, and link structure analysis. Temporal information was also used as a new feature to represent the Web images in order to meet the demand by users for the most recent information. In their case, the query is given as a multimodal object including an image and other kinds of information, such as a textual description. In constrast, our queries are composed of only image content and we want to search for objects in a multimodal collection of objects.

Calumby et al. (2012) addressed the problem of multimodal image retrieval by proposing the use of similarity functions that combine both visual and textual descriptions. In their approach, queries are defined in terms of both visual and textual evidences and GP is used as a learning technique that determines suitable combination functions based on available descriptions and user relevance feedbacks.

Such previous work has shown that GP is a suitable alternative for learning to rank in CBIR tasks. In this paper, we also adopt GP as our learning to rank method, although other alternatives can be explored in future work.

2.2. Related Work using Re-ranking Techniques

The work presented in (Arampatzis et al., 2011; Chen et al., 2010; Clinchant et al., 2011; Cui et al., 2008; Liu et al., 2009; Yao et al., 2010) exploits the relation between textual and visual pieces of evidence to improve results of image search. All these approaches share the idea of dividing the retrieval task into two main steps. First, textual evidence is used to obtain an initial ranking. Next, the visual patterns obtained from the initial ranking are used to perform a re-ranking of the results. Our work follows the inverse direction, being one of the first to explore such an approach. Visual evidence is used here to obtain the first ranking of results and then textual evidence associated with the retrieved images is used to perform the automatic re-ranking. This happens due to constraints in the target domain we deal in this article. It also brings new challenges since starting with the visual information may introduce some noise and uncertainty to the whole process due to the well-known semantic gap problem in CBIR, when compared to traditional textual retrieval. In order to deal with such problems, we use genetic programming to help finding the best possible query expansion and re-ranking possibilities in a huge space of possible solutions.

Besides the relevance feedback technique proposed in (Calumby et al., 2012) that also adopts GP, methods of visual re-ranking have also received increasing attention recently (Arampatzis et al., 2011; Jain and Varma, 2011; Liu et al., 2009; Pedronette and Torres, 2011; Popescu et al., 2009; Yao et al., 2010) in an attempt to improve the relevance of results in image search applications. Visual re-ranking can be defined as a technique to reorder the ranked documents of an initially given query. Such techniques are somewhat related to our proposal, since they exploit additional information in an original ranked list to improve effectiveness. Work in this area follows two distinct directions: (i) self-reranking or unsupervised re-ranking (Arampatzis et al., 2011; Liu et al., 2009; Pedronette and Torres, 2011; Popescu et al., 2009), which extracts information from the initial search results to automatically refine the ranking and reorder the results and (ii) example-reranking, which uses relevance feedback provided by users about the initial ranking to reorder the results. From now on, we focus revision of related work on the first direction, since our method can also be considered as a self-reranking proposal.

An unsupervised re-ranking approach which exploits the relation among images is presented in (Pedronette and Torres, 2011). The authors propose a method in which reference image collections are adopted to create associations between the image answers at query processing time, providing an information that they name as contextual information. The proposal is to analyze contextual informa-

tion considering the k-nearest neighbors to redefine distances among these neighbors with respect to other images of the collection. Based on the new distances, a re-ranking is performed and computed in an iterative way.

Following the same direction, the lightweight re-ranking method proposed in (Popescu et al., 2009) is based on the visual similarity between image search results and on their dissimilarity to an external class of diversified images. Some images of the external class are added to the query results in order to find out which elements are close to the class itself and far from the external class. The intuition is that relevant results are visually related to other answers to the same query and irrelevant results are near the images of the external class.

2.3. Related Work using Multimodal Information

A two-stage method for retrieving images from a multimodal collection is presented in (Arampatzis et al., 2011). The query is first processed based on textual information only. Next, CBIR is performed to re-rank the top-k items. The value of k is computed dynamically per query, ensuring that the CBIR will be performed on the better subset of the first ranking. Notice that authors address the problem of ranking results given a textual query, while we here address the problem of ranking results for an image query, which, as we have argued before, is potentially much harder.

In (Yao et al., 2010), it is assumed that there is a mutually reinforcing relationship between visual and textual features which can be reflected in the re-ranking procedure. The ranked list is organized into two connected graphs based on visual and textual descriptions. Each node in a graph carries a score based on the initial ranking. The method performs a random walk in this graph, assuming that the visual and textual consistent patterns are expected to receive higher scores.

A method called crowd-reranking, proposed in (Liu et al., 2009), aims at mining visual patterns which are relevant to a query from the search results of multiple search engines, again addressing the problem of ranking results for a textual query, while we here process an image query. Given a textual query, an initial ranked list of visual documents is obtained. Meanwhile, this query is fed to multiple image and video search engines. From the obtained results, it is detected a set of representative visual words by clustering the local features of image patches. These patterns are used to perform the re-ranking of the initial ranked list.

2.4. Related Work on Product Visual Search

The integration between visual and textual features to improve performance of product search for clothing and accessories is exploited in (Chen et al., 2010).

While we address the problem of improving visual search with multimodal features, the authors in that study exploit expansion of textual queries using visual features, the opposite multimodal expansion problem. They developed a text-guided weighting scheme for visual features. Such weighting scheme infers user intention from query terms and enhances the visual features that are significant towards such intention.

In a previous work, we addressed the problem of searching for products using an image as a query (dos Santos et al., 2013) and present a re-rank strategy which uses multimodal information to automatically re-order the original ranked list. As far as we know, this is the only previous work that addresses the problem of exploiting multimodal information to re-rank original results of a pure image query for product search. In this previous research, no learning technique was used in the ranking process and the proposed method requires several parameter settings which should be adjusted according to the collection, mostly in an *ad-hoc* way.

2.5. Baselines

As mentioned before, there is a vast literature related to image expansion using unimodal information (i.e., expanding images using other images (Wang et al., 2006; Rahman and Bhattacharya, 2009; Rahman et al., 2011),) as well as expanding textual queries with multimodal information, thus addressing a problem different from the one we deal here. Accordingly, we have decided to use our original proposal (dos Santos et al., 2013) as one of our main baselines for this current study, given that we were unable to find any other work that expands image queries with multimodal information for product search, i.e., our previous proposal is the only work in literature that provides a multimodal expansion for product image search when the query input is given as an image.

In this paper, we adopt the same dataset used in (dos Santos et al., 2013), which contains a collection with products, queries, and relevance judgments. Note that our previous proposal is an *ad-hoc* method with several parameters specifically adjusted to this collection. Our new GP framework, on the contrary, is much more flexible, presenting a complete method that can be easily adapted to other collections, since it adopts a generic (still very successful) learning process.

For sake of completeness's and to avoid doubts about the effectiveness our method, we also considered the introduction of a classic query expansion method, Total Recall, proposed by Ondrej Chum (Chum et al., 2007) in the baselines. While this method is considered as one of the state-of-art methods proposed in literature, it was not proposed to the type of application we address here, being

experimented before only in classic image collections for near-duplicate image detection. As it relies only on visual information to produce the final results, and given the multimodal nature of our target application, the method achieved a low performance in our experiments.

3. Visual Search in Visually-Oriented E-Commerce

The impact of the application of CBIR techniques in the visually-oriented e-commerce applications, such as the fashion domain, is potentially very high. In such domains, the use of only textual queries in order to search for a specific item usually leads to poor results due to the great generality of this domain. Moreover, an image example is usually much more descriptive than a textual query.

Sometimes, the use of an image to search for a particular product is essential in order to provide important stylistic details, which would be otherwise very difficult to express in a keyword-based query. This helps avoiding many false positives (e.g., stylistically different products) in the result set. For instance, the user may be interested in a specific tennis shoe found in a shopping center, and decided to search for similar tennis shoes in an e-commerce shopping by taking its picture.

3.1. Visual Features

There is a large number of image descriptors available in the literature each of which having its corresponding strengths and weaknesses. Image descriptors are used to characterize visual properties, such as color, shape, texture, or a combination of these properties, and encode such information into a feature vector representation. Once the image features are extracted, they can be used by CBIR systems to retrieve similar images to a given image query regarding the visual properties represented by the vector of features.

Since our problem is initially a CBIR task, we started our research by experimenting several image descriptors traditionally adopted in the literature. They are used to compute a distance measure between the image query and the images present in the dataset. Such distances will be adopted as terminals in our GP framework, as we shall see.

Some descriptors available in LIRE (Lux, 2011) were used in our experiments: CEDD (Chatzichristofis and Boutalis, 2008a), FCTH (Chatzichristofis and Boutalis, 2008b), CLD (Kasutani and Yamada, 2001), JCD (Chatzichristofis et al., 2009), ACC (Huang et al., 1997), and 128 RGB histogram (Lux, 2011). We also evaluated other descriptors which achieved competitive results in previous work

presented in the literature: BIC (Stehling et al., 2002); SDLC (Vidal et al., 2012); PHOG descriptor (Bosch et al., 2007), SIFT (Lowe, 1999) and CSIFT (Abdel-Hakim and Farag, 2006). We thus extracted features using a total of 11 distinct descriptors. To extract visual features we used LIRE (Lux, 2011), JFeatureLib (Graf, 2012) and VLFeat (Vedaldi and Fulkerson, 2008) packages.

3.2. Datasets

To evaluate the performance of the image descriptors, we adopted two image collections (dos Santos et al., 2013). The first one, named as *DafitiPosthaus*, contains $23{,}154$ products collected from two Web sites, Dafiti[2] and Posthaus[3], two popular on-line fashion stores in Brazil. The second collection, named as *Amazon*, contains $12{,}807$ images collected from Amazon[4], a worldwide on-line shopping.

In both collections, each product is represented by an image, a short textual description, and it is classified into a single category. The category information was extracted directly from e-commerce Web sites crawled to create the collection. In case of *DafitiPosthaus*, each product is classified into one of the following categories, as crawled from the web site: men's clothes, women's clothes, women's bags, women's wallets, men's wallets, backpacks, women's watches, men's watches, women's shoes, men's shoes, and belts. In case of the *Amazon* dataset, the products were classified according to the following categories: clothing and accessories – men, clothing, and accessories – women, shoes – men, shoes – women, and finally the category handbags.

We have included the *Amazon* collection in the experiments because it contains fewer categories when compared to *DafitiPosthaus*, and also fewer products, thus providing less diversity in products With the two collections we thus provide distinct scenarios to validate the methods.

The experiments were carried out with two groups of visual queries for both collections. The queries were obtained as follows:

Soft query set (soft) this query set is composed of 50 image queries randomly selected from e-commerce sites that were not present in our collections. The images are fashion product images with homogeneous background, which makes it easy to process the query.

[2] `http://www.dafiti.com.br` (As of 10/02/2012).

[3] `http://www.posthaus.com.br` (As of 10/02/2012).

[4] `http://www.amazon.com` (As of 03/20/2013)

Hard query set (hard) this query set is composed of 50 visual queries extracted from different sites of blogs, magazines, and newspapers. The images are, in general, pictures of celebrities using clothing or accessories of user interest. This sort of images represents a class of hard, but relevant queries for users searching for similar products. What makes this set of images harder to process is the presence of a considerable amount of noise information in the background.

We adopted these collections due to a lack of multimodal benchmark collections in the target (in this case, fashion) domain. The existing product image collections we have found available in the literature contain only images without any textual information associated with the products. Further, the usage of a collection from another domain would probably lead to non conclusive experiments or wrong conclusions, since the performance of image descriptors may vary according to the application. For instance, a descriptor that performs quite well for the Oxford collection, a quite common benchmark adopted in literature, was among the worst one in our application scenario, as we shall see.

The original collections already include a set of relevant products for each query used in the experiment, which were obtained by evaluating the top 25 results of a plethora of visual descriptors and evaluating results of query expansion methods experimented in the article where the collection was first presented.

For the experiments here reported, we included new relevance judgments covering all the descriptors adopted by us, since some of them were not included in our previous work (dos Santos et al., 2013), thus expanding the list of relevant answers. In order to evaluate relevance, we asked 30 volunteers to provide a binary relevance judgment for each answer and each query, thus providing a fair and impartial comparison in our experiments. The average number of relevant products per query is 55.5 and 35.5 in the soft scenario for *DafitiPosthaus* and *Amazon* collections, respectively. In the hard scenario, the average number of relevant is 15.2 and 22.8 for *DafitiPosthaus* and *Amazon*, respectively.

Finally, we have adopted two metrics to evaluate the methods in the experiments: P@10 and MAP (Baeza-Yates and Ribeiro-Neto, 2011).

3.3. Experiments with Visual Features

The results obtained by the experimented individual descriptors are depicted in Table 1. It shows that, in general, the best results for the collection and scenarios experimented were achieved by the CEDD descriptor in both collections. CEDD incorporates both color and texture in a histogram using a very compact

representation. We stress that these values are depicted just for a reference of quality of results among the individual descriptors in our scenario of experiments. Our focus however is not to compare these individual descriptors.

	DafitiPosthaus				Amazon			
	soft		hard		soft		hard	
	P@10	MAP	P@10	MAP	P@10	MAP	P@10	MAP
CEDD	0.496	**0.300**	**0.224**	**0.186**	**0.327**	**0.165**	**0.241**	0.125
FCTH	0.380	0.185	0.136	0.105	0.224	0.103	0.182	0.116
JCD	**0.516**	0.290	**0.224**	0.176	0.308	0.160	0.206	**0.144**
BIC	0.174	0.143	0.096	0.083	0.123	0.065	0.199	0.096
SDLC	0.088	0.070	0.020	0.015	0.197	0.053	0.053	0.019
PHOG	0.220	0.065	0.034	0.016	0.047	0.018	0.030	0.026
RGB	0.210	0.177	0.110	0.063	0.223	0.077	0.219	0.105
CLD	0.204	0.159	0.064	0.021	0.214	0.071	0.126	0.065
ACC	0.140	0.040	0.040	0.021	0.205	0.061	0.072	0.059
SIFT	0.156	0.073	0.020	0.006	0.132	0.044	0.020	0.018
CSIFT	0.300	0.105	0.042	0.022	0.148	0.050	0.076	0.025

Tabela 1: Performance of individual image descriptors in *DafitiPosthaus* and *Amazon* datasets. Higher values presented in bold.

A first observation that can be drawn from the experiments is that the individual descriptors had an overall poor performance in our target application. As explained in the introduction, this behavior was already expected given that related products may sometimes have distinct images. When analyzing these results, we can see that in our application the descriptors that obtained the best performance were the ones which combine texture and color information (CEDD, FCTH and JCD). Looking at the results, we realized that texture is particularly important, since it is common to find products with similar texture among the relevant answers. The descriptors based only on color information performed quite worse than this initial group (BIC, SDLC, RGB,CLD and ACC), but still we can see that color is also present as a selector of relevance for the users. Usually products with closer colors were marked as relevant and thus we decided to keep these descriptors in our experiments. The PHOG descriptor, the only one based on shape also achieved quite poor results. One explanation may be the large variations in perceptions and how the products are presented in the database. For instance, some clothes like shirts, are presented folded, while other ones are presented entirely opened. SIFT and Color SIFT (CSIFT) adopt a bag-of-words model based on

local features, also performed quite bad, being among the worse methods, while they are reported as one of the best descriptors for other image search applications. Previous references in literature reported this poor performance for SIFT and other methods based on bag-of-words model when applied to the task of searching products in (Shen et al., 2012).

The individual results obtained by these descriptors are quite poor. Given such results, we decided to investigate alternatives for improving the final query results for e-commerce product search tasks. We first investigated the possibility of combining the several individual descriptors to produce a single ranking function for image queries. When further investigating the problem, we realized that the multimodal information available on the product database collections could be a valuable source of information for improving the final ranking of results.

In the next section, we present the genetic programming framework we adopted as the alternative for using such multimodal information to derive functions for ranking results of visual queries.

4. Genetic Programming For Query Expansion

Genetic Programming (GP) is an evolutionary methodology introduced by (Koza, 1992). It is a problem-solving technique based on the principles of biological inheritance and evolution of individuals in a population. The search space of a problem, i.e., the space of all possible solutions to the problem, is searched by applying a set of operations that follow the theory of evolution, combining natural selection and genetic operations, to create more diverse and better performing individuals in subsequent generations with the aim of providing a way to find near-optimal solutions for a given task.

GP evolves a number of candidate solutions, called *individuals*, represented in memory as binary tree structures. Every internal node of the tree is a function and every leaf node, known as terminal, represents either a variable or a constant. The maximum number of available nodes of an individual is determined by the depth of the tree, which is defined before the evolution process begins.

An example of an individual represented by a tree structure is provided in Figure 1. In this example, an individual combines the values of three distinct features (F1, F2, and F3)[5], into a single score function. In the next sections, we present the features adopted in our proposal to compose individuals.

[5]A typical example of a feature Fn is the similarity value or distance between two images.

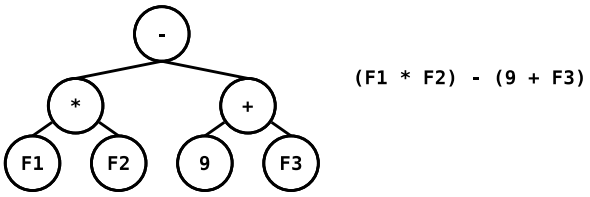

Figura 1: Example of a GP individual.

The evolution process starts with an *initial population* composed of a set of individuals. This initial population is generated randomly. Each individual is evaluated by a fitness function and associated with a fitness value. This fitness function is commonly modeled by a user-defined measure to score the ability of an individual to adapt to the environment (which in most cases correspond to the best solution for a given problem) and it is used to eliminate from the population all *unfit* individuals, selecting only those closer to the desired goal or those that achieve higher scores. In the case of search systems, this fitness function is the ranking quality measure we want to optimize. Individuals evolve generation by generation through genetic operations such as *reproduction*, *crossover*, and *mutation*.

Reproduction is the process that copies the "best" individuals from one generation to the next one without modifying them and that will also participate in the crossover and selection operations. The crossover operator allows genetic content exchange between two other individuals, the parents. In a GP process, two parent trees are selected according to a matching selection policy. Next, a random subtree is selected from each parent. The children trees result from the swap of the selected sub-trees between the parents.

Finally, the mutation operator has the role of keeping a minimum diversity level of individuals in a population. In the mutation operation, a random node in a tree is selected and then replaced by a new randomly created sub-tree.

Thus, at the end of the evolutionary process, a new population is created to replace the current one. The fitness value is measured for each new individual, and the process is repeated over many generations until the termination criterion has been satisfied. This criterion can be a pre-established maximum number of generations or some additional problem-specific success measure to be reached (e.g., an intended value of fitness for a specific individual).

The GP process adopted by us is the same adopted in (Mossri et al., 2007; da Costa Carvalho et al., 2012) for learning to rank. Its main steps are described in Listing 1, where we can see it is an iterative process with two main phases:

Listing 1: GP process adopted

```
1  Let $\mathcal{T}$ be a training set of image queries;
2  Let $\mathcal{V}$ be a validation set of image queries;
3  Let $N_g$ be the number of generations;
4  Let $N_b$ be the number of best individuals;
5  $\mathcal{P} \leftarrow$ Initial random population of individuals;
6  $\mathcal{B}_t \leftarrow \varnothing$;
7  For each generation $g$ of $N_g$ generations do {
8      $\mathcal{F}_t \leftarrow \varnothing$;
9   For each individual $i \in \mathcal{P}$ do
10         $\mathcal{F}_t \leftarrow \mathcal{F}_t \cup \{g, i, fitness(i, \mathcal{T})\}$;
11     $\mathcal{B}_t \leftarrow \mathcal{B}_t \cup getBestIndividuals(N_b, \mathcal{F}_t)$;
12     $\mathcal{P} \leftarrow applyGeneticOperations(\mathcal{P}, \mathcal{F}_t, \mathcal{B}_t, g)$;
13 }
14 $\mathcal{B}_v \leftarrow \varnothing$;
15 For each individual $i \in \mathcal{B}_t$ do
16     $\mathcal{B}_v \leftarrow \mathcal{B}_v \cup \{i, fitness(i, \mathcal{V})\}$;
17 BestIndividual $\leftarrow$ $applySelectionMethod(\mathcal{B}_t, \mathcal{B}_v)$;
```

training (Lines 5–13) and *validation* (Lines 14–16). For each phase, a distinct set of image queries is selected, which we call the *training set* and the *validation set*, respectively. It is important to stress that in this approach the whole GP process is performed offline, only once, when processing a training collection. After the process derives a ranking function, this (mathematical) function is applied to combine features at query processing time. For this reason, GP is an extremely low-cost learning-to-rank strategy method when considering run-time query processing computational costs.

The process starts with the creation of an initial random population of individuals (Line 5) that evolves generation by generation using genetic operations (reproduction, crossover, and mutation) (Line 12). The process continues until a stopping criterion is met. In the case of Listing 1, the criterion is the maximum number of generations of the evolutionary process.

In the training phase, a fitness function is applied to evaluate all individuals of each generation (Lines 9–10), so that only the fittest individuals are selected to continue evolving (Line 11).

In the case of learning to rank, each individual represents a weighting function that assigns a score to each image in the collection given an image query. The fitness of an individual corresponds to the quality of the ranking generated by the individual for each training query.

After the last generation is created, to avoid selecting individuals that work well in the training set but do not generalize for different image queries (a problem known as *over-fitting*), a validation phase is applied. In this phase, the fitness

function is also used, but at his time over the validation set of queries and documents (Lines 15–16). Individuals that perform the best in this phase are selected as the final solutions (Line 17).

4.1. Individuals

An individual is represented by functions and terminals, organized in a binary tree structure. As functions in the inner nodes, we use addition ($+$), multiplication ($*$), division ($/$), square root ($\sqrt{}$), minimum ($\min$), maximum ($\max$) and logarithm ($\log$). We use the genetic operators of reproduction, crossover, and mutation as proposed by (Mossri et al., 2007). Terminals, or leaves, contain information obtained from the sources of relevance evidence and constant values. We adopt the values in the range of [0..100] as constant. In addition to these, we also use values of the features related to the images and their expansions as terminals as follows.

4.1.1. Visual features as terminals

For each of the 11 individual descriptors presented in Section 3, we have an associated feature used as terminal in the GP process. The value of the terminals generated for each descriptor is a function that computes the distance between the image query and a given image from the collection, according to this descriptor. Further, we also adopted as features to each product the minimum distance score obtained when using the descriptor, thus adding 11 extra features. Each of these 11 extra features assigns an equal value to each image in the collection, however such type of constant value may be useful to the learning process. We have at the end a total of 22 visual features which represent terminal functions in the GP process. Besides these terminals containing visual information, we also expand the query to obtain multimodal features, as explained in the next Section. Using these features as terminals, it is possible to use the GP framework to combine information from multiple descriptors onto a single ranking function. We name this option as *Visual-GP* in our experiments and show that it can produce ranking results superior in quality when compared to the individual descriptors.

4.1.2. Obtaining multimodal terminals

In the product databases usually available in e-commerce services, it is possible to find not only images of products, but also other complementary important information, such as a textual description about each product and the category to which it belongs, since in these sites, products are often classified into categories. For example, an on-line store that sells clothes usually classifies its products into

men/women, shirts, dresses, shoes, etc. Further, each product found in such store will probably have an associated description.

A key question we investigate here is whether the learning-to-rank process can take advantage of such complementary information to enhance the quality of results when compared to systems that use only visual features to compute the final ranking. The first step to answer this question is to determine a way of associating multimodal information, such as category and textual descriptions with the image query. Our aim is to propose and evaluate strategies that do not require the user to provide any extra information about her information needs besides the image query. Thus, the association should be done without any explicit relevance feedback or complementary information provided by users. To solve this problem, we propose and investigate a total of 88 multimodal features to be automatically extracted from the image query. These features are extracted from two main sources, the category of images in the collection and the description of these images.

We have considered the usage of the expanded features in two modes. One uses the features to expand all documents from the collection, thus allowing new documents to be included in the top of the ranking. The second option is to perform a re-rank of the top results of the initial query, as we did in (dos Santos et al., 2013). Adding these multimodal features as terminals, we enable the GP framework to perform an automatic multimodal expansion of the initial image query submitted to the system. We now describe the multimodal features experimented by us to perform the multimodal expansion using GP.

Category of top similar images

In this strategy, we associate the image query with the categories found at the top answers of the visual results achieved by each of the visual descriptors described in Section 3. We compute for each descriptor the frequency of all categories of products found in the top answers. We then assign to each product in the collection the frequency of its category and we use that as a new feature to the learning process.

Figure 2 presents an example with an image query and its top five results according to a visual feature. When looking to the results, we see that four of the them are from the category "Clothing and Accessories - Woman" and one is from the category "Clothing and Accessories - Men". In this example, the result would create a feature terminal which assigns respectively values four and one to these categories, assigning zero to the remaining categories. Thus products that belong to those categories would be probably promoted in the ranking by this feature.

Query	Top-5 results	
		description: guess by marciano shauna mesh halter **dress** category: clothing and accessories - women
		description: kenneth cole women fluttery squares **dress** category: clothing and accessories - women
		description: tiana women fun pick up **dress** category: clothing and accessories - women
		description: london times women matte jersey mesh shutter tuck **dress** category: clothing and accessories - women
		description: rochester big and tall non iron pleated **shorts** category: clothing and accessories - men

Figura 2: Example of an image query and its top-5 results according to a visual feature.

We compute the frequency of categories found in a query taking the top elements into account for the expansion, considering the top with 1, 5, 10, and 20 elements. Notice that other number of element could be considered to compose the top in our framework, we however experimented this small set to check the effectiveness the proposed expansion strategy. The value of each terminal in this case is the frequency of the category of the product. Since we have considered 11 visual descriptors and 4 alternative ways of taking the top results, we thus produce a total of 44 new multimodal features with this strategy.

Text of top similar images

For each visual descriptor adopted, we compute a textual query with the concatenation of the textual descriptions of the top most similar products. In fact, only a segment of the text is adopted, since our previous work (dos Santos et al., 2013) indicated that taking just one or three terms of the product description is better than taking its whole description. In case of collections in Portuguese language, we take the first terms to represent products, in case of collections in English, we take the last terms of the collection. This difference is due to the structure of the languages, while in English the main subject of the description, the name of the product, usually appears at the end of the text, in Portuguese the first terms contain

this information. The segment size we adopted here was of size 1, meaning that we take the first or the last word of each product description when performing our multimodal expansion. This size was chosen because it gave good experimental results in our previous work (dos Santos et al., 2013) and also further reduces the computational overhead of the method.

The textual query obtained is compared to the description of each product in the collection, computing a similarity value between the query and each product. The similarity values are computed using the Vector Space Model (McGill and Salton, 1983). Those values are taken as features to the GP process.

Again Figure 2 can be used to illustrate this process. The top results for the visual queries in this case would be adopted to create a textual query. In this example, the textual query would contain four occurrences of the word "dress" and one occurrence of the word "short". This query would be submitted to the search system and the similarity of each product to this query would them be used as an extra source of relevance evidence when computing the final ranking.

Similar to the category information, we included the same 4 different alternatives to compose the top results, obtaining a total of 44 alternative new multimodal features.

4.2. Fitness Function

The fitness function must measure the quality of the ranking generated using a given individual. In our experiments, we adopted Precision at 10 (P@10) and Mean Average Precision (MAP) (Baeza-Yates and Ribeiro-Neto, 2011) as alternative fitness functions.

4.3. Selection of the Best Individuals

The validation step in our experiments was performed as proposed in (Mossri et al., 2007). According to this approach, the choice of the best individuals is accomplished by considering the average performance of an individual in both the training and validation sets, minus the standard deviation value of such performance. This method is called SUM_σ. The individual with the highest value of SUM_σ will be selected as the best.

More formally, let t_i be the training performance of an individual i, let v_i be the validation performance of this individual, and let σ_i be the corresponding standard deviation value of $(t_i + v_i)$. The best individual is selected by:

$$\underset{i}{argmax}((t_i + v_i) - \sigma_i) \tag{1}$$

Notice that a smaller value of σ_i makes a larger contribution to the selection, thus giving preference to individuals that have a more regular performance in the queries adopted in the training and validation sets, while $(t_i + v_i)$ also gives preference to those that perform well in both sets.

Finally, the whole GP process depends on the selection of an initial randomly selected seed. To reduce the possible risks of finding a low-performance local best individual, we adopted the same strategy proposed in (da Costa Carvalho et al., 2012), which consists of running N processes with distinct random seeds, and pick the best individual (according to SUM_σ) among those generated by these N runs. Previous results from the literature indicate that this strategy diminishes the chances of a single seed leading to a below-average performance, while it also avoids the experiments to obtain wrong conclusions by taking an outperforming seed by chance.

4.4. Multimodal Expansion Alternatives

We study four alternative ways of expanding queries with our GP framework. These alternatives are illustrated in Figure 3 and described as follows:

- *Expanded-GPI*: Our expansion method which processes the queries by adding textual and category features obtained from the results of each individual visual descriptor adopted. Figure 3(a) presents the main steps of *Expanded-GPI*, where we take as terminals all the visual features and simultaneously add the multimodal information extracted from them.

- *Expanded-GPC*: Our expansion method which extracts textual features from the results obtained from the combination of all visual features. In this case, we first apply the genetic programming method to achieve a good combination function for the visual descriptors, achieving an initial ranking (*Visual-GP*) , and then expand this initial ranking with the textual and category features. This alternative was considered to check whether the better initial ranking provided by *Visual-GP* could provide better expansion results. Figure 3(b) illustrates the main steps of *Expanded-GPC*. Notice that in this case, we run the GP process to obtain a combination function for the visual features and then extract the multimodal information from the result of this first run of GP. A second run is then performed to combine the multimodal features with the results of the *Visual-GP*.

- *Re-rank-GPI*: Similar to *Expanded-GPI*, but uses the textual and category features to re-rank the results from the union of the top 100 results found in

the visual features, instead of expanding the query answer set. In this case, we re-rank only the answers found in the union of the top 100 results of each individual visual feature, and results not present in this union are not re-ranked, while in *Expanded-GPI* a document not present in this union of top results may be inserted the final answer.

- *Re-rank-GPC*: Similar to the *Expanded-GPC*, but uses the textual and category features to re-rank the top 100 results of *Visual-GP*, instead of expanding the query answer set.

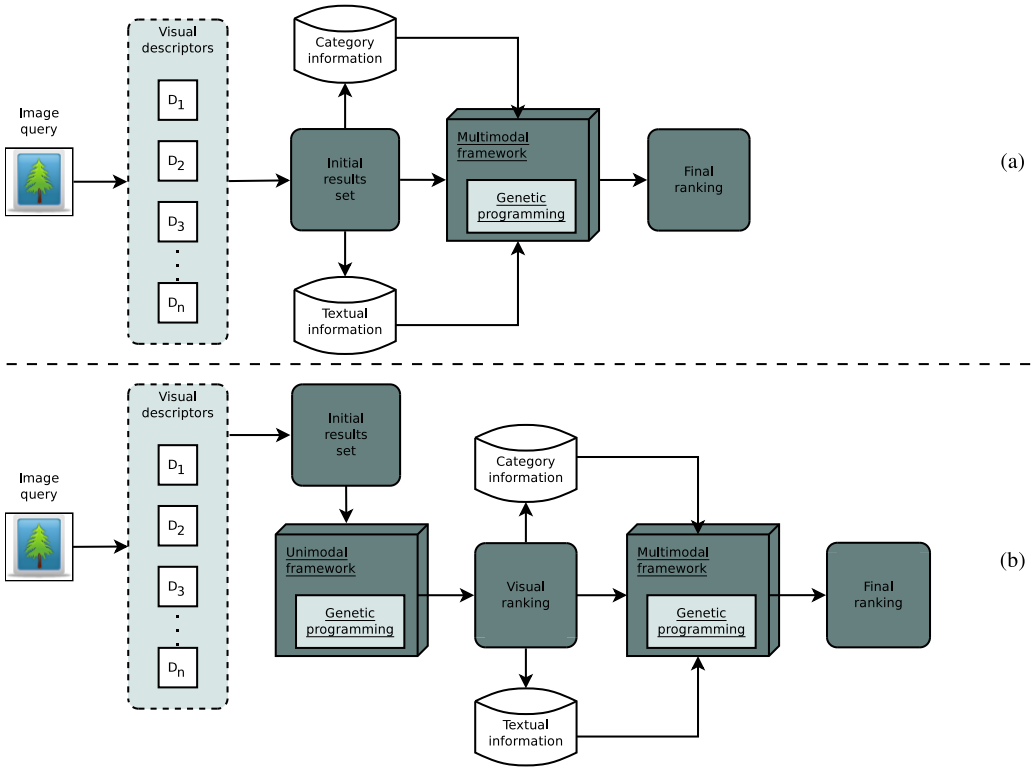

Figura 3: Overview of the *Expanded-GPI* (a) and *Expanded-GPC* (b) multimodal retrieval framework

While at a first glance this process may look expensive, it is important to stress that it is used to derive good combination functions, and thus it does not impact the query processing times, being a process to learn how to obtain good, and usually

not so expensive, combination functions to get the final ranking. Efficiency issues are better detailed in Section 5.3.

4.5. Adjusting GP Parameters

In order to set the GP parameters, we adopted the experimental design proposed by (Feldt and Nordin, 2000), performing a two-level full factorial design (Box et al., 1978) to investigate the impact of three parameters: the population size, the number of generations, and the maximum depth of an individual in the GP system. As a result, we concluded that a good setup for all GP processes experimented was to adopt a population size of 500 individuals, 40 generations, and a maximum depth of the generated trees was set to 7. For genetic operations, we used rates of 90%, 5%, and 5% for crossover, reproduction, and mutation, respectively. At the end of each generation, the validation phase was run for the top 20 best individuals returned by the training phase of that generation.

Finally, we adopted a 5-fold cross validation in the experiments. Each set of 50 queries was divided into 40% training queries, 40% validation queries, and 20% test queries for each fold. We executed this whole process 10 times and took the best individual from training and validation among them to apply at the test phase of each fold.

5. Experiments

In this section, we report the experiments performed to validate our multimodal learning-to-rank approach. We have adopted two metrics to evaluate the methods in the experiments: P@10 and MAP. We applied Student's *t-test* to check whether differences in the results obtained by each method are statistically significant, considering $p < 0.05$. Thus whenever we mention statistically significant differences in the experiments when comparing two methods, we mean differences with $p < 0.05$.

5.1. Baselines

We have included three distinct baselines for studying and asserting the impact of our proposal when compared to previous results, including two visual query expansion methods, one based on genetic programming which we named *Visual-GP* and the *Total Recall* (Chum et al., 2007) expansion method, one of the state-of-art solutions we found in literature for image search with query expansion.

Visual-GP is our genetic programming solution when applied only to the visual features described in Section 4.1.1. This comparison allows the reader to

better assess the impact of our proposed ideas of multimodal expansion in the final result. The parameters used in *visual-GP* were those described in Section 4.5.

As a second baseline we implemented the *Total Recall* method, an expansion method proposed in (Chum et al., 2007) that adopts the bag-of-visual-words architecture and performs a query expansion on the initial results. The image query is submitted and highly ranked documents verified by spatial constraints to suppress false positive images. Secondly, the verified images are used to learn a latent feature model to enable a controlled construction of expanded queries. We experimented *Total Recall* applying SIFT descriptor as proposed in (Chum et al., 2007), and also applying CSIFT descriptor, to compose the initial result. As expected, the best alternative for our problem application was the one using CSIFT, and we thus report only this version in the experiments.

We have also included the *ad-hoc* re-ranking method proposed in our previous work (dos Santos et al., 2013), *Term and Category-Based Re-ranking (TCatBR)*. It performs a multimodal re-ranking of the top k_1 results of an image descriptor (CEDD). It assigns the image query to the most frequent category found in the top k_2 retrieved images, takes the description of the top k_3 results which belong to this category and performs a textual query with the concatenation of them. The results of the textual similarities is then linearly combined with the visual scores to re-rank the top k_4 results. We adopted exactly the ones we found were the best for the two collections in our previous work, which are $k_1 = 100, k_2 = 20, k_3 = 20$, and $k_4 = 100$. We stress that we did not address a strategy for properly setting those parameters in our previous work. Thus, the GP framework proposed here is, as we show in our experiments, a viable alternative to implement the multimodal expansion with fewer *ad-hoc* parameter selection dependency.

5.2. Results

Table 2 presents the results obtained by our four alternatives of automatic expansion with GP, comparing them to the *Visual-GP* and *Total Recall* methods. Comparison to *Visual-GP* is important to assert whether our gain is due to the ability of GP to combine the several descriptors adopted or it is really a gain related to the addition of the expanded features. First, we can see here the advantage of using GP to combine visual descriptors in comparison the results presented in Table 1.

Visual-GP achieved higher scores than all individual descriptors. For instance, when considering P@10 in *DafitiPosthaus*, the best descriptor was JCD for soft and hard query sets. In both cases, *Visual-GP* presented statistically significant

gains over JCD. When considering MAP, in the soft and hard query sets, we obtained statistically significant gains of about 10% and 22%, respectivelly, over the best visual descriptor, CEDD. In sum, the usage of GP for combining features has obtained improvements when compared to the usage of any individual descriptor in all the scenarios.

Comparing the four alternative ways of using GP, the best performance was achieved by *Expanded-GPI*, which allows the expansion to include new results at the top answers and also obtains multimodal features from the ranking provided by each of the individual image descriptors. The options that perform re-rank of the top answers achieved gains when compared to *Visual-GP*, but performed worse than the expansion alternatives in most cases.

The strategies that expand the results based on the result of *Visual-GP* (*Expanded-GPC* and *Re-rank-GPC*) performed worse than the alternatives of expanding results from each individual descriptor (*Expanded-GPI* and *Re-rank-GPI*). That means the GP framework takes advantage of the information provided by each individual descriptor when computing the final ranking. The individual descriptors provide different ways of measuring the distance among the query image and the images from the collection. Such diversity allows finding related images according to distinct aspects, such as color, texture, and shapes. When expanding the queries from the individual descriptors, this diversity is then provided also in the several textual and categories obtained.

Expanded-GPI, our best expansion alternative, resulted in gains in all scenarios and in both collections when compared to *Visual-GP*, with differences statistically significant in all comparisons according to *t-test*. For instance, the gain obtained by *Expanded-GPI* when compared to *Visual-GP* was about 22.7% in the soft queries and 19.5% in the hard queries considering MAP for *DafitiPosthaus* dataset. When considering P@10, gains of *Expanded-GPI* were around 19.5% in the soft scenario and 37.6% in the hard scenario. Gains even higher were obtained in the experiments with the *Amazon* dataset, showing that neither the differences in number of categories nor the differences with regard the language between the two datasets affected the potential gains achieved when using our proposed expansion strategies.

When compared to *Total Recall*, all the expansion methods studied, including the usage of *Visual-GP* performed better than it. It is important to stress that *Total Recall* is an expansion method that is not based on learning, not designed to combine multiple features and was not proposed for the specific image search application we address in this article. Its inclusion in the experiments is important anyway to avoid doubts about its relative performance when compared to our

	DafitiPosthaus				Amazon			
	soft		hard		soft		hard	
	P@10	MAP	P@10	MAP	P@10	MAP	P@10	MAP
Total Recall	0,300	0,117	0,042	0,023	0.150	0.062	0.080	0.030
Visual-GP	0.604	0.330	0.250	0.220	0.342	0.168	0.259	0.147
Expanded-GPI	**0.722**†	0.405	**0.344**†	0.263	**0.419**†	**0.243**†	**0.367**†	**0.222**†
Expanded-GPC	0.678	0.367	0.285	0.232	0.394	0.221	0.327	0.196
Re-rank-GPI	0.704	**0.418**†	0.332	**0.276**†	0.391	0.226	0.315	0.187
Re-rank-GPC	0.694	0.391	0.313	0.258	0.371	0.209	0.293	0.173

Tabela 2: Performance of GP without expansion (*Visual-GP*), method *Total Recall* and the proposed expansion and re-rank methods based on GP when applied to *DafitiPosthaus* and *Amazon* collections. Higher values presented in bold, statistically significant differences between *Visual-GP* and the expansion methods are marked with (†).

proposal, being also useful to explicitate that general image search solutions may give poor results when compared to our methods specifically designed to search for products by using images.

Table 3 presents a comparison between the results obtained by the baseline *TCatBR*, which is our *ad-hoc* method published in a previous work and proposed specifically to deal with the two product collections adopted in the experiments, and the results with our best query expansion alternative using GP (*Expanded-GPI*). *Expanded-GPI* achieved higher scores in all collections, query scenarios and metrics. In *DafitiPosthaus*, the differences were statistically significant for MAP in soft and hard query sets with gains of 54.6% and 28.3%, respectively. When considering P@10, the gains were also statistically significant, being 12.8% and 19.4% for soft and hard query sets respectively. When considering results obtained with *Amazon* collection, gains were statistically significant for MAP in both query sets. For P@10, differences were statistically significant only in hard queries, although *Expanded-GPI* also obtained higher P@10 values in the soft query set.

The overall results indicate GP is a viable alternative to automatically expand the domain of image queries. Besides the competitive results obtained, the proposed GP expansion provides a framework which can be applied to other image search applications where multimodal information is available, while *TCatBR*, our previous approach, is a specific re-ranking method derived exclusively to be adopted to the search of fashion products.

A good point in the experiments is to understand what makes our expansion strategy work. To illustrate the advantages of our expansion method when com-

	DafitiPosthaus				Amazon			
	soft		hard		soft		hard	
	P@10	MAP	P@10	MAP	P@10	MAP	P@10	MAP
Expanded-GPI	**0.722**†	**0.405**†	**0.344**†	**0.263**†	**0.419**	**0.243**†	**0.367**†	**0.222**†
TCatBR	0.640	0.262	0.288	0.205	0.402	0.192	0.324	0.182

Tabela 3: Performance of the baseline *TCatBR*, our adhoc expansion method and the best proposed expansion method based on GP when applied to *DafitiPosthaus* and *Amazon* collections. Higher values presented in bold, statistically significant differences between *Expanded-GPI* and *TCatBR* are marked with (†).

pared to the method without expansion and to better understand its advantages in the results, Figures 4 and 5 show the P@10 difference between *Expanded-GPI* and *Visual-GP* for soft and hard queries, respectively, when using *DafitiPosthaus*. Checking individual query results, we realized that *Expanded-GPI* performed better for 64% queries, equal results in 14%, and worse in 22% if considering the soft queries. For the hard queries, our method achieved better results in 44% of the queries, equal results in 38% and worse results in only 18%.

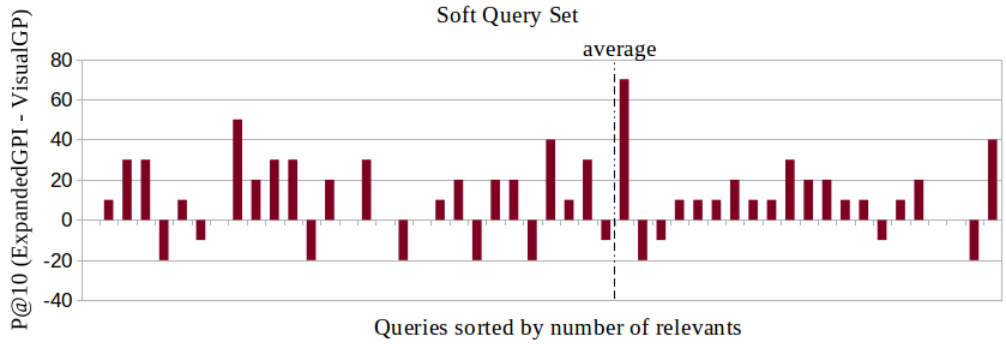

Figura 4: P@10 difference between *Expanded-GPI* and *Visual-GP* for each **soft** query when using *DafitiPosthaus* dataset.

When looking to the 28 queries where our method presented equal or worse results for the hard queries, we realized that they refer to the queries that contain fewer relevant answers, with 24 of them containing a number of relevant below the average presented in the dataset. The same phenomenon occurs when analyzing cases of losses in soft queries, where from the 11 queries where the expansion resulted in worse results, 7 are below the average number of relevant documents.

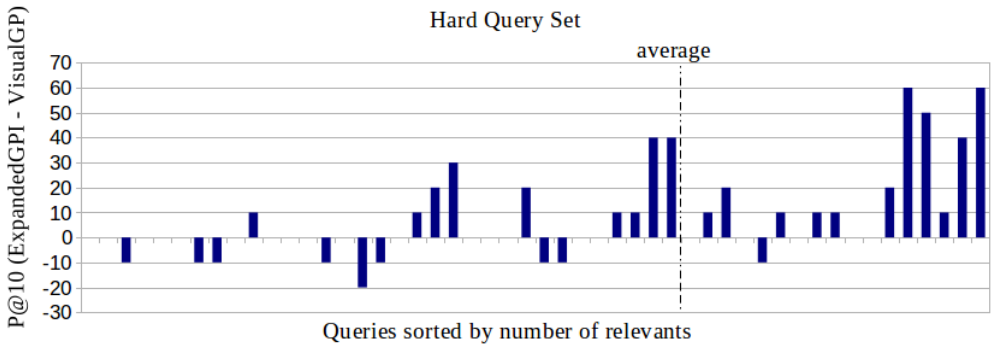

Figura 5: P@10 difference between *Expanded-GPI* and *Visual-GP* for each **hard** query when using *DafitiPosthaus* dataset.

To better illustrate the cases where we achieved improvements and also where the expansion resulted in loss, we present in Figure 6 an example of a query where our method resulted in a better ranking, according to the relevance judgment of the users. The image query presents a woman carrying a bag, which seems to be the searched product according to the relevance judgment of the users. As it can be seen, *Expanded-GPI* was able to filter out wrong results from the top.

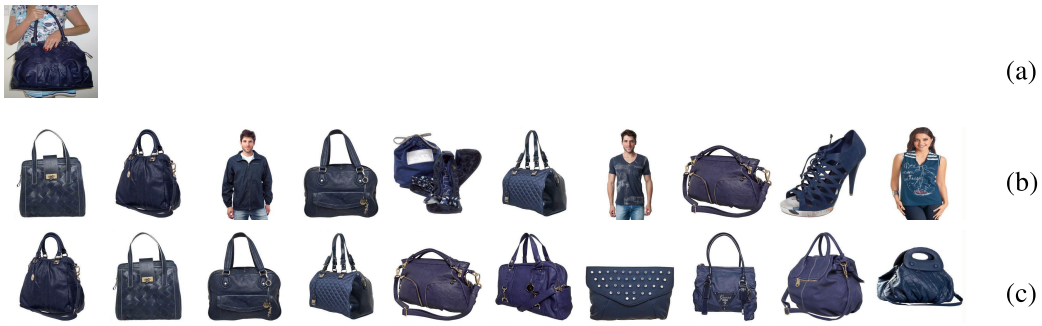

Figura 6: Examples of search results for image query (a), where *Expanded-GPI* (c) obtained better precision scores than *Visual-GP* (b).

Figure 7 shows an example where *Expanded-GPI* performed worse than *Visual-GP*. In this case, the image query is a men's shirt. The *Visual-GP* brought similar images to the query, but also presented a woman's shirt at the top 10 results. While the *Expanded-GPI* did not bring any woman's shirt at top 10 results, it included three t-shirts in the results that were not considered relevant by the users. Fur-

thermore these irrelevant answers were presented at the top of the results. This example shows that a minimum quality in the initial ranking is required to make the expansion work properly, this evidence is reinforced by the comparative graphics of Figures 4 and 5, which show the improvements are higher in queries that contain more relevant answers.

Figura 7: Examples of search results for image query (a), where *Expanded-GPI* (c) obtained worse precision scores than *Visual-GP* (b).

Examining the functions generated by our method, it is possible to see that descriptors that give good individual performance and the features derived from such descriptors usually have more influence in the final ranking. The generated functions encode quite large expressions. However, when analizing them, we see that descriptors such as CEDD, FCTH, CSIFT, ACC and JCD, and the features derived from them, usually appear with more evidence in the functions. For instance, Equation 2 presents a ranking function generated by the method *Expanded-GPI*:

$$csift_{text20} + fcth_{text10} - rlog10(max(fcth_{mindist}, rgb_{text1}) - (jcd_{mindist} + cedd)) + min(phog, jcd_{text20}) + phog_{text20} + rgb_{text1} + (cld_{text10} * cedd_{cat5}) \quad (2)$$

where the names of descriptors alone represent the distance between a document and the image query, according to this descriptor. A descriptor d, followed by text$n$ ($d_{text\{n\}}$), represents the textual similarity between the document and the textual description of the top n images in the initial ranking given by d. A descriptor d, followed by cat$n$ ($d_{cat\{n\}}$), represents the frequency of the category of the document in the top n results given by d. Function min returns the minimum value among its parameters. A descriptor d, followed by $mindist$ ($d_{mindist}$),

represents the minimum distance of any image in the data set to the image query, according to d. Looking to the example of Equation 2, we can see that the features derived from descriptors CSIFT, CEDD, FCTH, RGB, JCD, CLD and PHOG determine the final ranking in this sample function.

5.3. Computational Costs

Most of the computational costs related to our method are added in an offline process developed to derive the combination functions when using GP. When processing queries, the main overhead is to retrieve the multimodal features for the top results. In our experiments, we get this information for only up to the top 20 results. That means we need to retrieve the textual description of the top 20 results, create a query composed of the concatenation of these descriptions, and submit this textual query. Processing each of such textual queries is in fact fair faster than processing the initial image query, thus the final overhead for the process is not so high. In our experiments, this overhead was close to the time for processing the image query without any expansion. The final combination represents a very small cost and as a result the final overhead for processing queries in our experiments was about the same time for processing the initial visual queries. So, the final time was about two times the time for processing queries without our proposed expansion.

We stress that our experiments were performed using LIRE for processing the visual queries and Lucene for processing textual queries and all tasks were executed sequentially. In a real system designed taking efficient issues into account, this overhead could be further reduced. For instance, the expansion features for each visual descriptor could be taken in parallel, reducing the extra time for processing queries.

6. Conclusions and Future Work

The experiments carried out indicate that the idea of automatically expanding visual queries in CBIR systems is an interesting alternative to improve the quality of results when multimodal information related to the images is available.

We implemented and experimented the idea in the scenario of searching for fashion and accessories in e-commerce Web sites. This is an application that is becoming more relevant, given the growth in sales of this sort of products and the popularization of mobile devices capable of capturing images. The results obtained indicate that expressive gains in the quality of results can be obtained in this scenario when using our proposed multimodal expansion method. Compared

to ranking generated with a GP framework without expansion, we achieved gains of 22.7% in the soft queries set and of 19.5% in the hard query set in terms of MAP in *DafitiPosthaus* dataset. When considering P@10 measure, the gains are 19.5% in the soft scenario and 37.6% in the hard scenario when considering *DafitiPosthaus* collection. Gains achieved when using *Amazon* collection were even higher.

As future work, we intend to study alternative ways of selecting the initial sets of images taken into account when obtaining the multimodal features. We also plan to study the expansion strategy when using other learning-to-rank methods and also other image query scenarios. While we have experimented only one CBIR application here, the ideas may be applied to other scenarios, such as performing the multimodal expansion on large-scale web search systems based on CBIR. Finally, another research direction is to derive methods that automatically detect situations were the expansion may not result in improvements, thus avoiding unnecessary computational costs.

7. Acknowledgments

Authors thank the funding agencies and other institutions that contributed to this work.

Referências

Abdel-Hakim, A., Farag, A., 2006. Csift: A sift descriptor with color invariant characteristics. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Vol. 2. pp. 1978–1983.

Andrade, F. S. P., Almeida, J., Pedrini, H., da S. Torres, R., 2012. Fusion of local and global descriptors for content-based image and video retrieval. In: Iberoamerican Congress on Pattern Recognition. pp. 845–853.

Arampatzis, A., Zagoris, K., Chatzichristofis, S., 2011. Dynamic two-stage image retrieval from large multimodal databases. Advances in Information Retrieval, 326–337.

Baeza-Yates, R. A., Ribeiro-Neto, B. A., 2011. Modern Information Retrieval - the concepts and technology behind search, 2nd Edition. Pearson Education, England.

Bosch, A., Zisserman, A., Munoz, X., 2007. Representing shape with a spatial pyramid kernel. In: ACM CIVR. pp. 401–408.
URL http://doi.acm.org/10.1145/1282280.1282340

Box, G. E. P., Hunter, W. G., Hunter, J. S., 1978. Statistics for Experimenters: An Introduction to Design, Data Analysis, and Model Building, 1st Edition. John Wiley & Sons, New York, USA.

Calumby, R., da S. Torres, R., Gonçalves, M. A., 2012. Multimodal retrieval with relevance feedback based on genetic programming. Multimedia Tools and Applications, 1–29.
URL http://dx.doi.org/10.1007/s11042-012-1152-7

Chatzichristofis, S., Boutalis, Y., May 2008a. Cedd: color and edge directivity descriptor: a compact descriptor for image indexing and retrieval. In: ICVS. pp. 312–322.

Chatzichristofis, S., Boutalis, Y., 2008b. Fcth: Fuzzy color and texture histogram-a low level feature for accurate image retrieval. In: WIAMIS. pp. 191–196.

Chatzichristofis, S., Boutalis, Y., Lux, M., 2009. Selection of the proper compact composite descriptor for improving content based image retrieval. 6th IASTED International Conference 134643, 064.

Chen, Y., Yu, N., Luo, B., Chen, X., 2010. ilike: integrating visual and textual features for vertical search. In: ACM MM. pp. 221–230.

Chum, O., Philbin, J., Isard, J. S. M., Zisserman, A., 2007. Total recall: Automatic query expansion with a generative feature model for object retrieval. In: ICCV. pp. 1–8.

Clinchant, S., Ah-Pine, J., Csurka, G., 2011. Semantic combination of textual and visual information in multimedia retrieval. In: ACM ICMR. ACM, New York, NY, USA, pp. 44:1–44:8.

Cui, J., Wen, F., Tang, X., 2008. Real time google and live image search re-ranking. In: ACM MM. pp. 729–732.

da Costa Carvalho, A. L., Rossi, C., de Moura, E. S., da Silva, A. S., Fernandes, D., Jul. 2012. Lepref: Learn to precompute evidence fusion for efficient query evaluation. JASIST 63 (7), 1383–1397.
URL http://dx.doi.org/10.1002/asi.22665

dos Santos, J. M., Cavalcanti, J. M. B., Saraiva, P. C., de Moura, E. S., 2013. Multimodal re-ranking of product image search results. In: ECIR. pp. 62–73.

eMarketer, March 2012. Apparel Drives US Retail Ecommerce Sales Growth. http://www.emarketer.com/newsroom/index.php/apparel-drives-retail-ecommerce-sales-growth/, [Online; accessed 28-February-2013].

Faria, F. F., Veloso, A., Almeida, H. M., Valle, E., da S. Torres, R., Gonçalves, M. A., Jr., W. M., 2010. Learning to rank for content-based image retrieval. In: ACM MIR. pp. 285–294.

Feldt, R., Nordin, P., 2000. Using factorial experiments to evaluate the effect of genetic programming parameters. In: EuroGP. pp. 271–282.

Ferreira, C. D., dos Santos, J. A., da S. Torres, R., Gonçalves, M. A., Rezende, R. C., Fan, W., 2011. Relevance feedback based on genetic programming for image retrieval. Pattern Recognition Letters 32 (1), 27–37.

Graf, F., 2012. Jfeaturelib. [Online; Version 1.3.1].
URL https://JFeatureLib.googlecode.com

Huang, J., Kumar, S., Mitra, M., Zhu, W., Zabih, R., 1997. Image indexing using color correlograms. In: IEEE Computer Vision and Pattern Recognition. pp. 762–768.

Jain, V., Varma, M., 2011. Learning to re-rank: query-dependent image re-ranking using click data. In: WWW. pp. 277–286.

Kasutani, E., Yamada, A., 2001. The mpeg-7 color layout descriptor: a compact image feature description for high-speed image/video segment retrieval. In: ICIP. pp. 674–677.

Kherfi, M. L., Ziou, D., Bernardi, A., 2004. Image retrieval from the world wide web: Issues, techniques, and systems. ACM Computing Surveys 36 (1), 35–67.

Koza, J. R., 1992. Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge, MA, USA.

Liu, Y., Mei, T., Hua, X., 2009. Crowdreranking: exploring multiple search engines for visual search reranking. In: ACM SIGIR. pp. 500–507.

Liu, Y., Zhang, D., Lu, G., Ma, W.-Y., 2007. A survey of content-based image retrieval with high-level semantics. Pattern Recognition 40 (1), 262–282.

Lowe, D., 1999. Object recognition from local scale-invariant features. In: IEEE ICCV. Vol. 2. pp. 1150–1157.

Lux, M., 2011. Content based image retrieval with lire. In: ACM MM. pp. 735–738.

McGill, M., Salton, G., 1983. Introduction to Modern Information Retrieval. McGraw-Hill.

Mossri, H., Gonçalves, M. A., Cristo, M., Calado, P., 2007. A combined component approach for finding collection-adapted ranking functions based on genetic programming. In: ACM SIGIR. ACM, New York, NY, USA, pp. 399–406.

Pedronette, D., Torres, R., 2011. Exploiting contextual spaces for image reranking and rank aggregation. In: ACM ICMR. p. 13.

Piji, L., Jun, M., 2009. Learning to rank for web image retrieval based on genetic programming. In: IEEE IC-BNMT. pp. 137–142.

Popescu, A., Moëllic, P., Kanellos, I., Landais, R., 2009. Lightweight web image reranking. In: ACM MM. pp. 657–660.

Rahman, M. M., Antani, S. K., Thoma, G. R., Sep. 2011. A query expansion framework in image retrieval domain based on local and global analysis. Inf. Process. Manage. 47 (5), 676–691.

Rahman, M. M., Bhattacharya, P., 2009. Image retrieval with automatic query expansion based on local analysis in a semantical concept feature space. In: Proceedings of the ACM International Conference on Image and Video Retrieval. CIVR '09. ACM, New York, NY, USA, pp. 20:1–20:8.

Shen, X., Lin, Z., Brandt, J., Wu, Y., 2012. Mobile product image search by automatic query object extraction. In: ECCV. pp. 114–127.

Stehling, R., Nascimento, M., Falcão, A., 2002. A compact and efficient image retrieval approach based on border/interior pixel classification. In: ACM CIKM. pp. 102–109.

Torres, R., Falcão, A. X., Gonçalves, M. A., Papa, J. P., Zhang, B., Fan, W., Fox, E. A., 2009. A genetic programming framework for content-based image retrieval. Pattern Recognition 42 (2), 283–292.

Vedaldi, A., Fulkerson, B., 2008. VLFeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`.

Vidal, M., Cavalcanti, J. M. B., de Moura, E. S., da Silva, A. S., da S. Torres, R., 2012. Sorted dominant local color for searching large and heterogeneous image databases. In: ICPR. pp. 1960–1963.

Wang, X.-J., Ma, W.-Y., Li, X., 2006. Exploring statistical correlations for image retrieval. Multimedia Systems 11, 340–351.

Yao, T., Mei, T., Ngo, C., 2010. Co-reranking by mutual reinforcement for image search. In: ACM CIVR. pp. 34–41.