14

Review of Human-Computer Interaction Issues in Image Retrieval

Mohammed Lamine Kherfi
Department of Mathematics and Computer Science, Université du Québec à Trois-Rivières Canada

Abstract:

Image retrieval is an active area of research, which is growing very rapidly. Indeed, stimulated by the rapid growth in storage capacity and processing speed, the number of images in electronic collections and the World Wide Web has considerably increased over the last few years. However, with this abundance of information, people are continuously looking for tools that help them find the image(s) they are looking for within a reasonable amount of time. These tools are image retrieval engines.

When using an image retrieval engine, the user is continuously interacting with the machine. First, he[1] uses the system's interface to formulate a query that expresses his needs. Second, he provides feedback about the retrieved results at each search iteration. This allows the engine to provide more accurate results by using relevance feedback (RF) techniques. Third, he may be asked to assign a goodness score or weight to each image retrieved, which helps evaluating the system's performance.

In this chapter, we will review the main interactions between human and the machine in the context of image retrieval. We will address several issues, including:

Query formulation:

- How the user expresses his needs and what he is looking for
- The different ways the query can be formulated: keywords-based, sentence-based, query by example image, query by sketch, query by feature values, composite queries, etc.
- Query by region of interest (ROI) vs. global query.
- Queries with positive example only vs. queries with both positive and negative examples.
- Page zero problem: finding a good image to initiate a retrieval session.

Relevance feedback: we will try to answer questions like:

- Why do systems use relevance feedback?
- How can the user express his needs during the relevance feedback process
- How this information is exploited by the system to perform operations like feature selection or the identification of the sought image.

---

[1] Note that the masculine gender has been used strictly to facilitate reading, and is to be understood to include the feminine.

- The different families of RF techniques.
- Relevance feedback with retrieval memory, i.e., taking into account the value of old iteration queries when constructing the new one.
- Whether it is useful for the system to create user profiles, and the challenges it has to face.
- The number of RF iterations required to obtain satisfactory results.

Viewing retrieval results:
- Existing viewing techniques: 2D linear presentation, 3D-based presentation, etc.
- Different ways the resulting images may be ordered and presented to the user: similarity-ordered, time-ordered, event-ordered, etc.

Evaluation of the retrieval performance by the user:
- How the user can express his satisfaction/dissatisfaction about the retrieved images
- What about the ground truth in image retrieval evaluation?
- System response time and its influence on user satisfaction.
- The ease of use of the system's interface.

Other issues:
- User's needs: He may be looking for a specific image, for images that meet a given need (e.g. illustrate a concept) or simply browsing the collection looking for potentially "good" images.
- Etc.

1. Introduction

In the last two decades, the number of images in electronic collections has increased considerably. This is due to several factors, including:
- The substantial drop in prices of image acquisition devices. These devices include digital cameras, video cameras, cellular phones, surveillance cameras, scanners that can digitize analog images, etc. This drop in price has resulted in many people now owning these devices, which allows them to create personal collections. In addition, these images end up on Web pages and are thus available to the general public. Professional collections are no less substantial. For example, many museums have several hundred thousands of images representing their collections. Another example is the images used in medicine for different purposes, including learning, diagnosis and decision-making.
- The increase of storage capacity and lower prices for storage devices (hard disks, CDs, DVDs, external hard disks, etc.). Within only a few years, the size of a normal hard disk, for example, has gone from a few megabytes to several hundreds of gigabytes. Today, an ordinary user can have the space needed in his computer to store several millions of images.

In addition to this, and due to the development of new technologies, which allow to share images across the Internet and all types of networks, people can now access tons of images that were not accessible before.

This availability of information, however, created a new need that did not exist before: to find desired images within a reasonable time. This stimulated the emergence of a new area of research, which is currently rapidly developing, namely image retrieval. The main objective of this area of research is to develop tools that can help the user find the desired images within a reasonable time. These tools are generally called image retrieval engines, or image retrieval systems.

Different scenarios are possible for image retrieval. The most common scenario is the following:

1. The engine allows the user to create his query. It may be a text box in which the user enters keywords describing what he is searching. It may also involve a set of images from which the user can choose several as examples. Other ways of creating the query are also possible, as we will see later in the chapter.
2. The user creates his query.
3. The engine searches by comparing the query against the images in the collection.
4. The engine displays the resulting images for the user.
5. If the user is satisfied or simply wants to end the retrieval session, he stops. If not, he gives feedback about these results.
6. The engine uses this information and tries to find the most relevant results, and then moves to Step 4.

Human beings are at the centre of any image retrieval method since it is primarily their needs that the retrieval engine must cater to. In this way, the person who uses the services of a retrieval engine is in continuous interaction with it, and, at different stages: creating the query, examining the results, evaluating the engine, etc. The objective of this chapter is to provide an overview of the different steps during which the user interacts with the machine in the context of image retrieval. We should point out that this chapter is in no way a survey of existing image retrieval engines and retrieval techniques. The user interested by this type of survey can find a lot of good articles in the literature. For example, [1][2][3][4] and [5].

The chapter has been organized as follows: Section 2 discusses the different types of interactions between human and the engine, whereas Section 3 explores the different tools at the user's disposal for creating his query and the manner in which this query can be created. In Section 4, we will discuss similarity measures and their link with human judgement, and in Section 5, will focus on relevance feedback. In Section 6, we will delve into more detail about the different methods of viewing the results, and Section 7 covers engine performance evaluation. We will end the chapter with a short conclusion.

2. Interaction Modes Between the User and the Engine

User needs and the manner in which users search for images vary from person to person, and even for a given user at different times:

- Some users have a specific idea of what they are looking for, whereas others simply want to navigate through the database (DB) in search of an image that will catch their interest.
- Some users are looking for a single image whereas others are looking for several.
- Some are looking for a specific image (an image they have already seen), whereas others are looking for any image that could meet a given requirement (e.g. to illustrate a newspaper article).

Depending on the type of user and the user's needs, his way of interacting with the engine may vary. Two different methods of interaction can be identified, namely query-based search and browsing through a catalogue. For example, if the user is interested by a specific image, the search function may work best for him. If, however, the user does not have a

clear idea about what he is looking for, but simply wants to explore the DB to find potentially good images, browsing through a catalogue may be very useful. By drilling down in the catalogue, he can better pinpoint his needs and more accurately identify what he is looking for.
The first style of interaction, namely query-based search, can be summarized as follows. The user uses the engine interface to create his query. This query may be textual or visual as we will see in Section 3. A good interface must be easy to use, and must allow the user to express his needs (e.g. example images must be available). After the user has created his query, the engine searches through the DB to retrieve the corresponding images. This involves extracting features, calculating similarity measures between the query and images in the DB, possibly using an index, as well as sorting images based on similarity. Once the results are obtained, they are displayed to the user on the engine interface. A good engine must enable the user to give more details on what he is searching for, which helps the engine refine the results via what is called Relevance Feedback. All these aspects will be explained in detail in subsequent sections.
For the second method, browsing, the system starts by creating a catalogue by grouping similar images within a given class. This similarity can be calculated in terms of visual elements, semantic concepts, or both. It is best if the catalogue is hierarchical, which means that each theme at an upper level is subdivided into subthemes. Once the catalogue has been created, the user can browse the DB by starting with a theme, and then search by either drilling down through the sub-themes or moving across horizontally to other related themes. At any time he may decide to change theme or to simply end the browsing session.

3. Query Formulation:

As we mentioned above, machine-user interactions can be done through a query or by browsing through a catalogue. In the first case, the user must start by formulating a query, whereas the second method does not require a query. In this section, we will focus on the first scenario.
The first communication between the user and the image retrieval engine takes place when creating the query. Indeed, the engine needs to understand which image(s) the user needs in order to meet this requirement. Creating the query is a delicate problem and more difficult than it seems. Two questions arise at this point: 1) For the user, the challenge is to describe images that the user needs by using the few tools at his disposal; and 2) For the system, the challenge is to understand what the user wants based on the query he formulated. However, note that considerable advances have been made over the past few years, which facilitates interactions between the user and engine during the query formulation. In the rest of the section, we will look at the main existing techniques.

The user expresses his needs using text:
The first image retrieval engines used the same query formulation technique as the previous text retrieval engines. This technique involves allowing the user to provide a textual description of what he is looking for. The textual description can be either a group of keywords or a sentence.

Keyword queries:
The user expresses his needs by providing a keyword, such as in the following query: I am looking for an image that contains “an apple”. Most engines enable the user to provide

several keywords. An example of this type of query could be: I am looking for images that contain “oranges and apples”. When the query is made up of several keywords, they may be combined using different logical connectors, such as AND, OR and NOT. This method of formulating queries is directly derived from text search techniques.

Query by sentence:
For this type of query, the user provides a sentence that describes what he is looking for. An example of this type of query could be: I am looking for “an image in which people are eating in a park”. The challenge with this type of query is to analyze the sentence in order to extract the most important words to the user. Another challenge involves understanding the exact meaning of the sentence, since a sentence is not a simple group of words without any order or links. For example, the word “Impala” can have different meaning depending on the sentence in which it is found. If the user creates the query “Find me a herd of impala”, the engine must understand that he is talking about the animal. However, if he says: “Find me a Chevrolet Impala on the road”, the engine must interpret it as the car, and not the animal.

Discussion:
Creating the query using a textual description presents a certain number of advantages:

- This is a natural way of allowing the user to express himself as he does in everyday life.
- It allows to re-use an entire arsenal of text-search techniques, which were developed over the years.
- It was noted by several researchers that text more easily captures semantic concepts associated with images. Imagine, for example, a user who is searching for images describing the concept “Joy”. As we will see a little later, the current content-based search techniques have great difficulty in extracting this concept from images automatically. If text is used, however, it becomes entirely possible to answer the query provided that certain images are annotated with this word.

This being said, a text-based search is not without its problems:

- First of all, this technique becomes unusable when the collection does not contain any text along with the images. This is unfortunately the case for most personal image collections. People often do not take the time to add text to their personal photos. Many of them just empty out their acquisition devices (cameras, etc.) by recopying the images onto their hard disk. This is also the case for many professional collections.
- Secondly, even if the images are annotated with text, this annotation can be very subjective. The same image can be annotated with different words by different annotators. According to [6], the annotation tells us more about the annotator than about the image itself.
- The text depends on the language. In order to be able to search in a DB in which the images were annotated by using a given language, there must be tools that translate queries into other languages to the annotation language.
- If images are surrounded by text, such as on a Web page, this text may be used in their indexing. This technique is used by certain retrieval engines on the Web. The problem however arises from the fact that it is not easy to determine which words are relevant to the image, and which words are not.

- Text does not go beyond a certain degree of refinement. For example, we went to Google Image [53] and searched using the word "Goose". We found the images in Fig. 1. on the first page of results.

Figure 1. Search results using the word "Goose"

All these images do indeed contain geese. What happens now if you are interested in images containing geese, but that must also visually resemble of the image in Fig. 2? This image contains a single goose, in a very specific position, with very specific wings and colours, water of a given colour and texture, etc. It is impossible to describe all these details using text, which demonstrates the limitation of the capacity of text to go beyond a certain level of refinement. We will look at how searches using an example image can get round this obstacle.

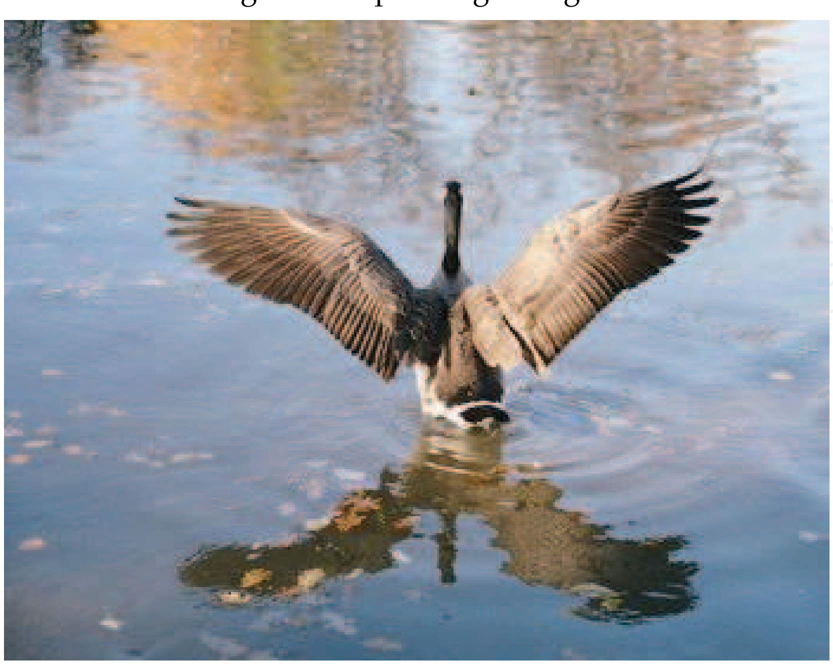


Figure 2. [54] Illustration of the limitation of text to describe the content of an image

- A picture is worth a thousand words: It can contain many objects with a given layout, very specific colour shades and shapes that cannot be described with text. Take the image in Fig. 3, for example. It contains houses that are shaped in a specific way, cars of specific makes, models and colours, trees, lawn, poles, etc. All these objects are set up in a particular way. How could we describe the entire content of this image in words?

Figure 3. [55] A picture is worth a thousand words

The user expresses his needs using images:
The limitations of text-based retrieval that we have mentioned earlier have led certain researchers to wonder whether it would be better to let the images speak for themselves. In other words, the idea was to allow the user to formulate his queries using images, and then the system would quite simply find the images that resemble them. Of course, responding to these queries that only contain images means that different techniques must be used than with textual queries. This new method was called content-based image retrieval or CBIR. As part of content-based searches, the query can be formulated in different ways, which we will summarize below. However, note that a certain number of steps are common to most methods:

1. A certain number of visual descriptors must be extracted from all the DB images. This extraction must be done a priori, i.e., before even allow the user to perform searches.
2. In general, the same visual descriptors must be extracted from the query.
3. The comparison between the query and a DB image comes down to comparing between their visual descriptors.

Query in which the user provides the value of each feature
Some engines, such as [7], have chosen this technique, which involves asking the user to provide the numerical value of each feature. If, for example, each image is described by colour moments and Fourier descriptors of its shapes, it is then up to the user to provide the numerical value for these features. It is clear that creating this type of query is, for various reasons, very difficult, if not impossible, for the user, even if he is a specialist in image processing. First of all, the ordinary user ignores the meaning of features, such as colour

moments or Fourier descriptors. Secondly, it is extremely difficult, even for a specialist, to translate one's needs (the image that he is searching for) into a set of numerical values.

Query based on example image(s)
This is definitely the most successful content-based technique. The principle of this technique is simple: the user selects an example image, and then the engine finds images that resemble it. Several variations have been proposed:

Query with one example image versus query with several example images:
In its simplest form, retrieval by example image can be summarized as follows:
1. The engine starts by proposing to the user a number of images from the DB.
2. The user selects one of these images to say "Find me images that look like this".
3. The engine browses through the DB looking for the images that resemble the query, and then returns the results to the user.

The images that the engine proposes at the beginning may be chosen randomly or intelligently (e.g. an image from each family). In general, wisely choosing these images can make the search easier, more productive and quicker.
Many engines allow the user to create a query with several example images. In this case, the example images can be combined by using logical connectors, such as AND, OR, NOT, etc. At the time of the search, the images that make up the query may be combined in different ways and at different levels. They can be combined at feature level by calculating, for example, an average of all these images, and then by comparing this average with the images in the DB. The combination can also be made using set operations. If the user is looking for images that resemble Image A AND Image B, then the engine can start by searching all images that resemble A, and then all images that resemble B, and the result will be the intersection of the two sets.
Later we will see that certain engines make use of the fact that the user chooses several example images to perform feature selection. However, note that certain models need many example images to be able to select features. This can be restrictive and requires much work on the part of the user, which is not always guaranteed.

Query by global image versus query by region of interest:
Certain retrieval engines do not allow the user to select part of an image as the query. If an image is selected, it is taken as a whole. However, it was noted in different situations that the user may be interested in part of an image, instead of the entire image. An example of this is the user who is looking for a given object regardless of the background on which it appears. In this case, allowing the user to select part of the image as a query can considerably improve search results.
Searching by regions of interest can be summarized as follows:
- During feature extraction phase, each image is first segmented into regions; then each region is represented by a set of descriptors.
- The user creates his query by selecting one or more example regions. Certain engines require the regions to be chosen within a single image whereas others allow to select regions from different images. In addition, certain engines enable the user to choose regions as negative examples.
- Once the query has been created, the engine searches through the collection images for those that can be described by the combination of example regions.

- When it involves comparing a single region with another, different similarity measures can be applied, including probabilistic measures and distances. The problem becomes more complex, however, when comparing two groups of regions: the first coming from the query, while the second comes from the DB image. To perform this comparison, different techniques were adopted, including fuzzy logic [8] and set operations.

Lastly, note that nothing prevents the retrieval engine from letting the user to combine global images and regions of interest in the same query. A possible example would be "I am looking for images that resemble Image I but without Object O."

Queries with positive example versus queries with negative example:
Over the past decade, researchers have realized the importance of negative examples and the additional possibilities these offered for creating queries. The negative example can let the user express what he does not want, which helps solve several problems during image retrieval, including noise and miss. Noise is the set of images that the user does not want, but that are returned by the engine. Miss designates all images that should have been returned, but were not. There can be different reasons for these two problems: a user who did not express his needs well, an engine that was not successful in understanding these needs, etc. The negative example can be used as a way of reducing noise and miss. By selecting a few images as negative examples, the user tells the engine that these must be skipped (as well as any image resembling them) in the results of the next iteration, which reduce the amount of noise. As well, the images skipped will be replaced by more relevant images, which reduces miss.
Some of the other advantages of negative examples include:

- It allows to target certain parts of the search space that the positive example alone cannot do. It also finds classes of results that have complex forms in the search space.
- It can sometimes solve the Page zero problem, which we will discuss at the end of this section
- It helps better select features.

Many engines enable the user to combine positive examples with the negative examples when formulating the query. Some engines allow users to introduce the negative example since the first iteration whereas others allow users to use it only for the second iteration, i.e., to refine the results. Technically, the negative example was modelled in different ways, including optimization models [9], probabilistic models [10] and set models [11].
Note that the negative example alone does not really allow to create a query due to its multi-modality. Indeed: if you know what someone does not want, this does not give you enough indication to know what he wants. For example, if a user does not want images of cars, this can mean that he is looking for trees, buildings, the sea, grass, works of art or anything else.

Queries by sketch and queries by predefined icons
Two other mechanisms for creating the query have been adopted by certain engines. The first involves allowing the user to make a sketch that roughly represents what he is looking for, such as in [12]. The user can use the mouse or a electronic pen to make the sketch. As well, the engine can allow the user to colour the objects being drawn.
The second mechanism involves letting the user "make" his own query image. The engine starts by proposing a list of icons, each representing a well-defined object, such as the sky, sea, sun, a car, etc. The user can select icons that interest him and put them in the right place

on a "canvas", which represents his query image. The engine of [12] allows this type of query.
These two ways of creating the query can be useful in certain situations:

- They sometimes help in the case of a *target search*, i.e., a user who searches for a specific image that he has already seen. However, in the case of a *group search* or when the user does not have a specific description of the image content that he is looking for (e.g. he wants to find any image that can illustrate a given concept), these types of queries may be unsuitable.
- They allow to create certain simple queries (e.g. an object on a background). However, they do not allow to describe complex images, such as images with a multitude of objects.
- They can help solve the Page zero problem. When the user cannot find the right image to start the search from those proposed by the engine, the sketch or icons can help serve as a starting point. He starts by making a sketch or placing several icons, and then the engine searches for a few corresponding images, and lastly the user uses some of these images to create his query.

However, these types of queries are fraught with a number of problems:

- They depend largely on the ability of the user to express his needs by using the sketch, which is not easy given the difficulty that some users have with sketching, especially with a mouse or a electronic pen.
- Many users do not have the time or patience needed to place icons or draw a sketch for each search iteration.
- Another difficulty crops up when searching and comparing since the engine must compare two things that are not of the same type: a sketch and an image. One possible solution consists of comparing the shape of the drawn objects or selected icons with the shapes extracted from the DB image. Another possible solution would be using shape and object recognition. Automatic annotation could also be used.

Given their limitations, these two types of queries cannot be used alone in an engine. They must be combined with other methods, such as query by example images.

Discussion:
Content-based retrieval includes a certain number of advantages, including:

- The fact that it can be used even if the DB does not contain any text. Indeed, in this case, the text-based search becomes unusable, and the only way would be to base the search on the content of the images.
- It works well with very complex images and with those containing many objects that cannot be described with text.
- It allows a level of refinement that text cannot. For example, looking for images that visually resemble the image in Fig. 2 is quite possible using a search with example images.
- The content of images is more objective than text.

Content-based search also have a certain number of challenges:

- Extraction of visual features.
- Semantic gap.
- The page zero problem.

Each of these challenges is discussed below:

Extraction of visual features:
The fact of designing and extracting visual features which accurately represents the content of images is perhaps the pillar of content-based search. A multitude of features are proposed in the literature. They can be grouped in different families. The first family describes the colour and includes histograms, moments, etc. The second family describes the texture and includes the co-occurrence matrix, Gabor filter, autocovariance, etc. The third family describes the shape: this includes invariant moments, Fourier descriptors, edge points, etc. The fourth family involves mixed features that describe more than one aspect, such as the correlogram, which describes both colour and texture. Other features were also proposed to describe the structure, points of interest, etc. Extraction of features is a problem that is not completely resolved and much work remains to be done, especially regarding features that can capture the semantic content of images.

Semantic gap:
Although it works well for users interested in the visual content of images, content-based search have much difficulty in capturing semantics. For example, imagine a user who is searching for images that can be associated with the concept "Lunch". A search on the Web using Google Image [53] provides the results in Fig. 4. Although these images describe this concept, there is little or no visual resemblance between them. How, then, can the content-based search meet this query? This lack of connection between the visual content of an image and the semantic concepts that may be associated with it is known as the Semantic Gap.
Different solutions were proposed to alleviate this problem. Some simply combine the content of images with text, since text better captures semantics. Others use relevance feedback in order to better understand what the user wants. However, note that the problem of semantic gap is far from resolved. It is the greatest challenge that the new generation of content-based retrieval engines face.

Figure 4. Results of the search using the word "Lunch"

Page zero problem:
It sometimes happens that none of the images proposed by the engine resemble what the user is looking for, and therefore cannot be used to create the query. This is known as the

page zero problem. Several solutions can be applied to solve or alleviate this problem. Certain engines allow the user to select another set of images from the DB, which can serve as examples. Other engines allow the user to provide his own example image, i.e., an image that is not in the DB. However, this should not be the only option possible seeing that the user does not always have the images to describe what he wants. Queries containing several example images can also provide some solutions to the page zero problem, insomuch as each of these images contains part of what the user wants. Queries by region can also be useful. For example, imagine that, from the images being proposed by the engine, only one contains the object that the user wants, but that this same image contains other objects that the user does not want. In this case, forcing the user to choose the integrality of the image is restrictive whereas allowing him to select only the object that interests him provides more flexibility. The negative example can also contribute to solve the problem of page zero. As we explained earlier, the negative example reduces miss; and, when miss is reduced, the odds will be greater that the user finds new images that resemble what he is looking for. These images can therefore be used to create the query, which allows to overcome the page zero problem. Another possible solution for this problem is to start with a textual query and then refine it using example images, assuming of course that the engine supports textual queries. Lastly, note that the queries made using sketches and icons sometimes help solve this problem, as we explained earlier.

Combining different types of queries

Each way of creating the query is better suited for a given type of search, and meets a specific need. Text-based search allows to find images based on their semantics. Content-based search allows finding them based on their visual content, and is indispensable in the case of non-annotated DBs. As well, specific method of creating queries, such as sketches and icons, allow to solve certain problems, including the page zero problem. We think that combining all these types of queries in the same engine could only be an advantage. More tools would be available to the user, which would help him better express his needs. A possible scenario would be to conduct a two-step search. During the first step, the text is used to limit the search space to the set of images that relate to the same theme as the query. During the second step, the visual aspect is used to refine the results and sort them according to their visual resemblance with the query.

4. Similarity and Human Judgement

In the case of text-based search, matching techniques are generally used to compare terms contained in the query and those accompanying the images. If, however, the search is based on content, similarity techniques are more appropriate since rigid matching does not work in most situations. Indeed, requiring that an image be an exact match to the query to be returned to the user is a very restrictive choice and may return no result. Even images that are very similar to the query are almost never an exact copy of it. This is due to a certain number of variations and imperfections: difference in scale, angle, position, and object orientation, etc. Unlike matching, similarity does not require equality among images. It simply involves calculating a level of resemblance between the query and each image in the DB, and then sorting these images in decreasing order based on this degree of resemblance.

In image retrieval, a good similarity measure must be as close as possible to human judgement. This is justified by the fact that, in the end, it is the need of a person that should

be met, and it is him who will judge whether the results are relevant or irrelevant. Because of this, similarity is a complex cognitive process that involves different disciplines, including psychology, mathematics and computers.
Old models consider similarity as a distance in the feature space, which assumes that it meets the following conditions: non-negativity, identity of indiscernables, symmetry and triangular inequality [13]. However, experimental studies have shown that these conditions are not always met.
The *Thurstone* and *Shepard* models, where the base idea comes from [13], represent a second family of similarity models. These models can be seen as a generalization of distances. See [14] for a good review of these methods. In these models, the similarity between two stimuli (images in our case) is a function of the distance, which is Minkowski's distance, given by $d(x,y)=\left[\sum_{k=1}^{n}\left|x_k-y_k\right|^{\gamma}\right]^{\frac{1}{\gamma}}$.
Later, other models were developed. These models drop the distance model, which allows them to eliminate the conditions mentioned earlier. We can talk about the work of Amos Tversky [15], which proposed the famous *feature contrast model*. Instead of considering stimuli as points in the metric space, Tversky characterizes them as a set of features. Let us assume that a and b are two stimuli, and that A and B are their respective sets of features. The work of Tversky stipulates that similarity can be obtained by calculating a linear combination of functions of common features ($A \cap B$) and discriminatory features ($A - B$) and ($B - A$). Mathematically, the similarity can be formulated as follows: $S(a,b) = f(A \cap B) - \alpha f(A - B) - \beta f(B - A)$, where f is a positive function and α and β are two constants.
Some work, including [16], noted that all stimuli do not influence the perception of similarity according to the same mechanism. For some of them, a distance may be appropriate and correspond to test results. Others, however, require more complex models.
Before ending this section, we would like to say a few words on the similarity between colours, because colour is a feature that is largely used when searching for images. Colour can be characterized in different ways including histograms and colour moments. As well, it can be represented in different spaces: RGB, HSV, XYZ or even L*a*b*. It was noted that certain spaces correspond better to human judgement than others. For a space to be considered close to human judgement, the following conditions must be established: two colours that are distinct to humans must be found far from one another within this space, and two colours that are similar to humans must be close to one another within this space. For example, the L*a*b* space was often used since it approached human judgement. Lastly, for similarity measures used with histograms, different measures were used, including Euclidian distance [9], the Earth Mover Distance (EMD) [17] and histogram intersection [18].

5. The User and Relevance Feedback (RF):

Problem: Why do we need RF?
The user who interacts with an image retrieval engine expresses his needs through query formulation. However, due to imperfections at different levels, it is not unusual for the user to not be able to express his needs correctly or the engine to not succeed in understanding these needs. Different problems may lie at the origin of this lack of understanding between the user and engine. First of all, there is the semantic gap. Often the user is interested by the

semantics of images (e.g. I am looking for images that illustrate joy), whereas the engine relies on their visual content. The opposite may also occur: a user interested by the visual content of images versus an engine that only takes into account the semantic concepts extracted from the text, for example. Secondly, there is the weakness of the visual features to correctly represent the images. In spite of the progress made these past few years in feature extraction, a lot of work remains to be done before we can rely on features to adequately represent the content of images and even less so their semantics. Thirdly, there is the disparity between the similarity measures used by the engine and human judgement of the similarity between the images. The page zero problem is the fourth problem. It occurs when no image proposed by the engine resembles what the user is looking for, and cannot therefore be used as an example image. Fifth, there is the subjectivity of the text in the representation of images. The user of the engine and the person who annotated the images do not necessarily have the same point of interest, which means they will not use the same terms to describe the same image. Consequently, at the time of carrying out the search, the user will have a lot of problems finding this image. Other difficulties also crop up when we use text: synonyms, dependence on language and culture, etc.

Relevance feedback as a solution:
Relevance Feedback (RF) was introduced as a technique to overcome or alleviate the aforementioned problems. RF was first used in search techniques in the mid-sixties. Its objective is to improve retrieval precision during the iterations, based on the information the user provides about the relevance of the retrieved results. The first work on RF includes [19], [20] and [21]. Motivaed by the improvement it achieved in text retrieval, image retrieval researchers very quickly understood the role that RF could play in image retrieval, and have integrated it into their engines.

How does the user express his needs during the RF process?
The concept of RF is to ask the user to provide feedback regarding the results returned by the engine at each iteration. Using this mecanism, the user explicitly or implicitly provides more information on the images he likes and those he does not like as well as on the features that interest him and those that do not.
In concrete terms, RF can be carried out in different ways:

- The engine can ask the user to choose from the images returned at each iteration the ones that he finds relevant (positive examples) and the ones he finds not relevant (negative examples). It can also ask the user to assign a weight to each image. For a positive example image, a high weight means that it resembles very much what the user is looking for, whereas a low weight means that it resembles the user's idea a little. For a negative example image, a low weight means that similar images would not be appreciated, whereas a high weight means that the user definitely does not want similar images returned.
- The engine can ask the user to explicitly assign a weight to each feature used. However, this can be restrictive given that the normal user ignores the significance of features. In addition, even for a specialist, it is difficult to say whether a given feature is important or not to find what he is looking for. To resolve this problem, many engines guess the importance of features without explicitly asking the user. This information can be deduced from the example images that the user provides, as we will see in this section.

- Some engines ask the user to choose between the use of textual features, visual features or a combination of both.

What can RF do?

After the user has provided his feedback about the results of an iteration, this information is used to improve the results in different ways [10][5]. It helps to understand what the user is looking for, i.e., to identify the image(s) in his head. It also helps determine the importance he gives to each feature, which will then be used to define the similarity measures that best reflect his judgement.

The different RF techniques:

Early CBIR systems that adopted RF were built on the vector model in information retrieval theory. They used the query-point movement technique, and/or the axis re-weighting technique [22]. In the query-point movement technique, the ideal query point is moved toward the positive example and away from the negative example. Examples of systems that have adopted this technique include [23] and [24]. Rocchio's formula [25] has been frequently used to perform query-point movement. In the axis re-weighting technique, the main goal is to assign more importance to features according to which example images are close to each other, and less importance to other features. This can be justified by the fact that, if the variance of the query images is high along a given axis, any value on this axis is apparently acceptable to the user, and therefore this axis should be given a low weight, and vice versa [22]. An example of axis re-weighting models can be found in [23], where each feature is weighted with the inverse of its standard deviation.

More recently, some researchers have considered RF to be a classification problem in which example images provided by the user are employed to train a classifier, which is then used to classify the database into images that are relevant to the query and those that are not. Bayesian models have been used in systems like [26] and [27], which support image classes that assign a high membership probability to positive example images and penalize classes that assign a high membership probability to negative example images. SVMs have also been used in RF [28] [5]. Examples include [29] and [30]. Some systems first train an SVM classifier using positive and negative examples, and then use it to divide the database into relevant images and irrelevant ones. Considering RF as a classification problem may entail some difficulties, however. First, in a typical classification problem, each item (image) belongs to one or more clearly defined classes, whereas, in image retrieval, human subjectivity makes it difficult to assign a given image to a given class [31]. Second, classification does not always provide a ranking of the retrieved images in terms of their resemblance to the query, which may be necessary for some applications.

Other researchers consider RF to be a learning problem in which examples fed back by the user are used to train a model, which is then used for retrieval. Techniques used include self-organizing maps (SOMs), Bayesian frameworks and decision trees. In [32] for example, SOMs are used to measure similarity between images. In [33], a Bayesian framework is used to predict what target image users want, given the action they undertook. In turn, [34] proposes an RF model that, for each retrieval iteration, learns a decision tree to uncover a common thread uniting all images marked as relevant. This tree is then used as a model for inferring which of the unseen images the user would most likely want. The initial drawback of learning methods is the lack of data. Indeed, users usually provide a small number of feedback images in the retrieval process, while these algorithms need a large number of examples for training. For example, after extensive experimentation with the system described in [9], we found that people rarely give more than a few images as feedback, while the model, in order to be trained correctly, needs a number of images at least equal to the dimension of the largest feature. It would be inconceivable to ask the user to select several dozen images in each retrieval step, because this can make the retrieval process very slow and cumbersome.

Some researchers considered RF to be a distance optimization problem whose solutions are the parameters that make it possible to find the ideal query, weight the features, and transform the feature space into a new one that corresponds better to the user. Examples of such models include [9], [22] and [35]. In these models, RF is formulated as a minimization problem whose solutions are the optimal query and a weight matrix, which is used to define a generalized ellipsoid distance as a measure of similarity between images. The basic idea of those models is to enhance features for which example images are close to each other. When the query embeds some negative examples, they enhance features that distinguish clearly between positive and negative examples, and neglect those that do not. Like learning techniques, optimization techniques suffer from the problem of lack of data, and different attempts have been made to address it, like in [36], where the authors introduce the regularization method and the null-space method.

Type of user and user influence on RF strategy:
For relevance feedback, two different strategies can be used [28]. The first strategy is the most common. It involves providing the user, at each iteration, with the most relevant images that the engine was able to identify. The second strategy involves returning the more informative images and trying to obtain as much information as possible from the user. This helps to better pinpoint the range or set of images that the user is searching for. In [37] and [38] for example, at each iteration, two images are presented to the user, who must choose the one that most matches what he is looking for.

The difference between the two strategies is that the first one assumes that the user is impatient and therefore must be provided with the best results as quickly as possible or else he could end the retrieval session. The second technique assumes that the user will cooperate [39]. It attempts to ask him as many questions as possible to learn more about what he is looking for. Note that both techniques can be combined in the same system [40]. For example, at each iteration, the system can provide the user with the most relevant images and ask him some optional questions (if he wants to answer them) to better understand what he wants.

Relevance feedback with or without memory:
When processing a given iteration query, some RF models take into account older queries (previous iterations), whereas others only look at the query of the current iteration. The first family could be called “models with memory” and the second “models without memory”. Models with memory assume that the user is consistent in his choices, i.e., that, during a given session, he continues to search for the same images and does not change his intention. Models without memory do not make this hypothesis. Therefore, when the user is consistent, the precision of models with memory increases throughout the iterations. Some studies, including [41], have noted, however, that the user often changes his intention while searching. In this case, a model without memory may be the most appropriate. Technically,

models with memory consider the new search target as a combination (linear or otherwise) of the very last query and the queries from previous iterations.

Creating user profiles:
The concept of memory discussed in the previous subsection can be expanded even further. The engine can, for example, try to create a profile for each user. It must first identify each user in a unique manner. This can be done by asking him to identify himself each time he uses the engine by entering his user name and password for example. Other techniques, such as IP address or cookies, also can be used to identify the user or his machine. The second step is to memorize the preferences of each user when he performs search. The third step consists of using these preferences in the future to improve search precision. Let's take an example. User X created Query Q at a given moment. According to the his feedback, the engine understands that he was satisfied with the results obtained. In the future, if this same user submits the same query, it would be intelligent on the part of the engine to return the same results. However, if the user is not satisfied, the same results should not be returned. User preferences go beyond the set of resulting images. The user may have a preference for a given feature versus others, a type of query (e.g. text-based) versus others (e.g. content-based), etc. All this information can be stored by the engine for future use. Once individual profiles have been created, the engine can make a classification in order to discover the different user classes and preferences of the members of each class. This classification can be cross-referenced with their other attributes: age, sex, language, culture, etc.
Lastly, we should note that the creation of profile poses a certain number of challenges. The largest challenge is the potential cooperation of the users: to create profiles, users must be willing to identify themselves or provide certain personal information. This sometimes goes against protecting the user's privacy.

Helping the user create his query and provide feedback:
Sometime it is best to guide the user throughout the search process: from query formulation to relevance feedback to obtaining results. From the user's point of view, this assistance can make search easier and more attractive. The engine, in turn, can better understand the user's needs to better serve him.
This help can be provided in different ways:

- Help the user choose the query mode that best suits him from those offered by the engine: textual query, example image query, etc.
- Provide the user with some tips for creating his query, as is done by the engines of [42] and [43].
- When the engine asks the user to enter the importance he gives to each feature, explain to him at least the meaning of each of these features.
- Step by step and by asking a certain number of questions, the engine can have the user express his need in a more specific manner. For example, for the first step, the engine may propose a set of image families (animals, cars, landscapes, etc.) and ask the user to choose the family that corresponds to his search. Once the user has made his choice, the engine proposes the list of subfamilies, and so forth, until the desired results have been obtained.
- Guide the user, as in [37] and [38], where, at each step, the engine asks the user to choose the image that best meets what he is looking for between two images proposed.

- An “Advanced Search” function, found in certain Web retrieval engines, can be very useful. It enables the user to give more details about what he is looking for: file format (jpg, gif, bmp, etc.), file size, image dimensions, greyscale or colour, a photo versus a sketch versus a synthesized image, etc.
- Hints that appear automatically when the mouse moves over certain elements in the interface.
- Add a “What is it?” button beside certain elements in the interface so that the user can, if he so wishes, better understand their meanings.

While helping the user is definitely appreciated, we must however determine how far this help can go before it produces a negative effect. In extreme cases, we could require the user to take training so that he can benefit from all engine functionalities. However, we must remember that many users do not have the desire, patience or time to take this training. It therefore becomes an obstacle that purely and simply pushes them to abandon such an engine.

6. Results Visualisation:

1D Visualisation versus several D Visualisation:

Once the search has been performed, the engine must display the results to the user. The most used and traditional method is to present the results linearly with images ordered based on their resemblance to the query, starting with the closest match. We can call this way of presenting results the one-dimensional method. However, we should note that most engines use the fact that the screen is two-dimensional (2-D) and present the results on several lines where the first line has the most relevant images from left to right, as though reading a book.

Other methods of viewing search results have been proposed:

1. The system can use two features—which may be multidimensional—to represent images. All images in the DB are displayed in a 2-D plan, with the query image in the centre. Both axes of the plane each represent a feature. The position of a DB image on each axis is proportional to its dissimilarity to the query with regard to the feature concerned. In [44], for example, both axes represent the RGB and HSV histograms respectively. The X-axis of each DB Image I is obtained from the intersection of the RGB histograms of I and the query, with a positive sign if the entropy of the RGB histogram of I is lower than the entropy of the RGB histogram of the query. The Y-axis is calculated in the same way, but by using the HSV histograms.
2. Since most retrieval engines use more than two features to represent images, the method described in 1) cannot be used in a 2-D plan. It can, however, be generalized as follows: start by representing the images in the multidimensional feature space, and then project them in a 2-D space (plan). It is this plan that the user will see displayed, with his query in the middle. In order to minimize loss of information due to projection, techniques such as Principal Component Analysis (PCA) can be applied. This method was used in [45]. It not only allows to display images based on their similarity to the query, but also based on resemblance between them.
3. Some engines, such as [46], visualize images in a 3D virtual reality space. The three axes can each represent a feature as in 1) or a combination of features after projection, as in 2). In [46], for example, the axes represent colour, texture and structure

respectively. The engine can enable the user to view the results based on each axis taken individually, or even view them from any angle (combination of axes).

4. In certain engines, such as [47], the query image is displayed in the middle, and then surrounded by similar images. The size and position (distance) of each image from the query is proportional to its similarity to the query. In addition, [47] proposes two ways of displaying these images: either in concentric rings or in a spiral.
5. Some engines, such as [48], use Self-organized maps (SOM) for viewing collections of images.

Note that some of these methods can also be used when formulating the query. They can also be seen as a hybrid solution between the query and navigation.

All these viewing methods can be improved by combining them with the following techniques:

- Image size: during display, the size of each image can be in proportion to its similarity to the query.
- Zoom function: enable the user to zoom in to see more detail of a part of the collection or to zoom out to have a more global view.
- Reduce overlapping: When projecting, many similar images can be found in the same small zone, which means that some of them hide others. This effect is known as overlapping. This problem becomes even more serious when the collection contains many images. Most of the time, it is not possible to eliminate overlapping completely. However, it can be reduced by using optimization or heuristic algorithms, which attempt to find the position of each image that is as close as possible to its original position and that minimizes overlapping with its neighbours. Displaying images in small sizes with the Zoom option also alleviates this problem. However, the images should not be too small, since users will not appreciate this. The article, [49], analyzes and proposes a few solutions to these problems.

Size of image displayed to the user:

Whether during query creation, relevance feedback or results display, images must be displayed to the user. The issue that this section looks at is the choice of dimensions for the images displayed: Should the actual dimensions be kept, or modified, and why? The most natural choice would be to have each image keep its actual dimensions. However, this may have certain disadvantages:

- The actual dimensions of an image may be very large. The interface may not be able to display them. As well, displaying a large image may require greater calculation capacity and therefore take lots of memory.
- The different images can have different sizes. It is neither convenient nor attractive to present images of different sizes on the same interface.

Instead, the dimensions of each image should be adapted to the interface. We could replace each image with a thumbnail, for example, while giving the user the option of viewing the original image if he so wishes. The size of the thumbnail should be proportional to the original dimensions in order not to distort the image. In general, a thumbnail is smaller than the original image. However, thumbnails should not be too small. If they are, the user will have to view the original image each time to be able to see the details and decide whether the image interests him. This could be cumbersome and slow down the search process [41].

Presentation order of results:

Another issue that should be raised is: In which order should results appear? There are several possible solutions. The most common is to present them in decreasing order of similarity to the query. Other solutions are also possible:

- In chronological order according to creation date.
- By event: images taken during a given event are presented together. This could be family events or otherwise. An example of an event could be "Our camping trip in 1998" or even "Wedding of X family member". This way of presenting assumes that we know the event related to each image. It works well with certain collections of personal or family images.
- Hierarchically: different classes of images are displayed to the user, who can then choose the class that he wants to visit. This way of presenting the results is similar to browsing through a catalogue.
- A combination of all these choices.

Number of images to return to the user and interrogation technique:

Another issue that must be addressed is identifying the number of images to return to the user and how to find these images. Two main query techniques were used by most retrieval engines: The k nearest neighbours and the neighbours whose distance from the query (or dissimilarity) is below a certain threshold ε. If we use the first technique, a certain number of problems must be addressed. The first is choosing the number k. This choice however is not necessary for small DBs. The engine can simply sort all the images in the DB according to their resemblance to the query, and then return the first ones to the user, while giving him the option of viewing more results. For large DBs, using an index becomes essential. If this index is available, the search will be limited to the classes closest to the query, which leads us to searching in a smaller DB as in the previous case. The second problem is that the images returned may not resemble the query, especially when the number of relevant images in the DB is low.

When the second technique is used, the value of ε must first be determined. In order for the results to have meaning, a threshold under which all the images actually resemble the query must be chosen. Sometimes we have to define a variable threshold that changes depending on the query. The problem with the threshold technique is that it depends considerably on Recall. If it is too low, it might not return any results, and if it is high, it could return too many results. In the latter case, the results can be truncated by limiting them to the k nearest neighbours to the query, which brings us back to the first technique. The engine can also sort the results, display the first ones to the user and give him the option of viewing others.

Regardless of the query technique adopted, using an index is only useful when the DB is very large. An appropriate indexing technique limits the search to the most relevant classes, which helps increase precision of results and reduce search time.

Viewing on small devices and adapting:

These past few years, the use of portable devices (PDA, cell phones, Palm Pilot, etc.) has increased substantially. These tools are used for various purposes, including browsing the Internet, accessing multimedia collections and searching on the Web. Creating retrieval engines for these devices or adapting existing engines to them will help meet a growing need. Recently, some researchers have taken an interest in this issue. For example, [50], [51] and [52].

When a retrieval engine is developed for these devices, client-server applications can be used, in which the server runs on the computer to allow access to and searching in the DB, while the client runs on the portable device. Portable devices are different from a conventional computer. They are subject to additional limitations. The first constraint is the size of their screens, which are smaller than a regular computer screen. Therefore, the client interface and the size of images displayed must be adapted based on to this limitation. For example, the client program can display a single image at a time, but while giving the user the option of scrolling through the page to see more images. The second constraint is the reduced data transfer speed. The server must therefore limit as much as possible the number of images and data sent to the client. The third constraint is their rather limited calculation capacity. The server must perform a maximum number of operations, leaving the client only with the simplest things to do, like displaying results.

7. Users as a Retrieval Engine Evaluator

A retrieval engine is created to meet the needs of the user. The user must therefore be satisfied with the services offered by the engine. According to Section 2, there are two types of services: query-based search and catalogue browsing. In this section, we will look at issues related to evaluating each of these services.

Evaluating the search function:
The most common evaluation scenario is the following. We start with several retrieval sessions by changing the query each time. Once results have been obtained, they are evaluated by being assigned scores as to their relevance versus the query. These scores can be assigned by humans or obtained from preclassification of the DB. The scores of the different sessions are then combined, for example, using a weighted average, which allows to obtain different performance indicators, including Precision and Recall.
Therefore, it can be deduced that evaluating the search function of an engine requires three components, namely, an image DB, ground truth and evaluation measurements, as detailed below.

Image collection or DB:
In order to ensure an objective evaluation, the image collection used must meet a certain number of criteria. First it must be large enough to allow evaluation of the scalability of the engine. Next it must correspond to the objectives for which the engine was designed. If the engine was developed for personal photos, for example, it must be evaluated on a collection of personal photos, and if it was developed for art images, it must be evaluated on a collection of art images. While remaining within the engine's domain, the collection must be as diverse as possible in order to evaluate the ability of the engine to find images from different categories. Lastly, note that several collections have been used to evaluate search engines. The most commonly used is that from Corel [37].

Ground truth:
Ground truth allows to judge whether the images returned by the engine are relevant or not. Two types of ground truth are generally used: human judgement and preclassified DBs.
Human judgement is a good indicator, because, in the end, it is humans that will be using the engine. If they see the results as being relevant, then we can say that the engine is

precise. In order to ensure that the evaluation by people is objective, we must follow a certain number of recommendations:

- The number of users: A significant number of users must participate in the evaluation process in order to limit the effect of subjectivity from certain people.
- The users must be representative of the population who will be using the engine: the level of expertise in the field, level of instruction, age, sex, preferences, etc. For example, if the engine is for the general public, the evaluators should not all be experts in image processing.
- Sometimes training is required so that a person can use the engine. This training must be as concise and easy as possible; if not, users might not use the engine.

When the ground truth comes from a prior classification of the DB, a high score is automatically assigned to any image belonging to the same class as the query, whereas a low score, zero or even a negative score, is assigned to images from other classes. However, similar classes must be monitored: an image from a class resembling the query should not be considered as poor, even if it is not as good as an image coming from the same class as the query.

Particular attention must be given to preclassification. The fact of relying on preclassification to evaluate the engine implicitly assumes that it is perfect: imprecise preclassification would completely distort our evaluation. Preclassification can be obtained in different ways. It can be carried out by humans, which brings us back to the first type of ground truth, or it could be done by the machine with little or no human intervention.

Evaluation measurements:

Two of the most commonly used measurements are Precision (*Pr*) and Recall (*Re*). Precision measures the proportion of good images versus the total number of images returned to the user. Recall measures the proportion of good images returned to the user among all good images in the DB. Noise, which is the opposite of Precision, was also used. It represents the proportion of irrelevant images from all images returned to the user. Certain variations of Precision and Recall take into account image rank: the most relevant images must appear in the first positions. Once calculated, Precision and Recall can be represented by curves. Some authors draw the curve $Pr = f(Re)$. A good system should provide high Precision regardless of the Recall value. However, if Recall is low, as in the case of certain image collections, this measurement becomes unsuitable. Other authors replace it with $Pr = f(\mathrm{Sc})$, where Scope Sc is the number of images returned to the user.

Evaluation of the browsing function:

In order to provide browsing services to users, the engine must start by indexing the DB. This operation involves dividing the DB into classes, and then dividing each class into subclasses. The first thing to evaluate here is the class quality. A good class must be coherent and complete. Coherence means that the images assigned to this class resemble each other. An example of an incoherent class would be a class that contains images of apples, cars and horses. Completeness means that we find, in a given class, all the images that should be assigned to it. We can draw a parallel between Coherence and Precision on the one hand, and Completeness and Recall on the other.

As in the case for search, to evaluate the cataloguing, a DB is needed on which the algorithm will be applied and a ground truth can be used to judge relevance. The collection must be carefully selected. As for the ground truth, it can be provided by humans.

The catalogue can be evaluated based on the total number of images it contains, the diversity of subjects it covers, whether it is hierarchical or not, the inter-class and inter-level relationships, the option of moving from theme to theme, ease of use, etc. Some of these measurements, such as the total number of images covered by the catalogue, are objective. They can be calculated without requiring any judgement from the user.

Other evaluation criteria
A certain number of other criteria could also be used when evaluating:

Number of images indexed:
It is more difficult, but more useful, to index several hundreds of thousands of images than to index only a few dozens. An engine can be evaluated based on the number of images it indexes.

User-friendliness:
One of the attributes that makes a retrieval engine successful is how easy it is to use. The interface must be user-friendly on all levels: formulating the query, displaying results, relevance feedback, etc.

Response time:
Response time can also influence user satisfaction. A system, even if it provides relatively precise results, that takes too long will not be appreciated by users. The response time for a query can be influenced by:

- Prior extraction of features: extracting the features of images first means large time savings, since the engine will not have to do it at the time of the search. The only thing left for it to do is compare the query with the DB images. All known engines extract features ahead of time.
- The number of features used and their sizes: increasing the number of features or increasing the size of a few features generally increases the comparison time, and, in turn, the response time.
- Similarity measures used: certain similarity measures are quick to calculate whereas others take longer, which directly affects searching time.
- The fact of using an index: the index restricts the search space to classes that most resemble the query, which considerably reduces the searching time.

Refinement and number of iterations:
It is always a good idea for the engine to provide the user with the option of refining the results via relevance feedback. However, the number of iterations required to obtain good results should be minimal.

8. Conclusion

When a user uses an image retrieval engine, he is in constant interaction with the engine, be it to create the query, provide feedback, view the results or evaluate engine performance. In this chapter, we have looked at most of these aspects. The retrieval engine must meet human needs. Therefore, it must be as close as possible to them. In particular, it must use the features that capture the semantic content of images, use similarity measures that resemble human judgment, have a user-friendly interface, etc. A lot of work has been done to that effect over the past few years; however, we believe much remains to be done.

9. References

R. Datta, D. Joshi, J. Li and J. Z. Wang. Image Retrieval: Ideas, Influences, and Trends of the New Age. *ACM Computing Surveys*, Vol. 40, No. 2, 2008 [1]

A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta and Ramesh Jain. Content-Based Image Retrieval at the End of the Early Years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 12, pp. 1349-1380, 2000. [2]

M. L. Kherfi, D. Ziou and A. Bernardi. Image Retrieval from the World Wide Web: Issues, Techniques, and Systems. *ACM Computing Surveys*. Volume 36, No 1, pp. 35 – 67, March 2004. [3]

R. C. Veltkamp and M. Tanase. Content-Based Image Retrieval Systems: A Survey. *Technical Report* UU-CS-2000-34, Department of Computing Science, Utrecht University, October 28, 2002. [4]

Y. Liu, D. Zhang, G. Lu and W.-Y. Ma. A Survey of Content-Based Image Retrieval with High-Level Semantics. *Pattern Recognition*, Vol 40, N 1, pp. 262 – 282, 2007[5]

R. Jain. World-Wide Maze. *IEEE MultiMedia*, Vol 2, N 2, 1995. [6]

G. Wei, D. LI, and I. K. Sethi. Web-WISE: Compressed Image Retrieval over the Web. In *IAPR International Workshop on Multimedia Information Analysis and Retrieval*, 33–46. 1998. [7]

Jia Li James Z. Wang Gio Wiederhold. IRM: Integrated Region Matching for Image Retrieval. *ACM Multimedia* 2000. 147-156. [8]

M. L. Kherfi, D. Ziou and A. Bernardi. Combining Positive and Negative Examples in Relevance Feedback for Content-based Image Retrieval. *Journal of Visual Communication and Image Representation*, Vol 14, pp. 428–457, 2003. [9]

M. L. Kherfi and D. Ziou. Relevance Feedback for CBIR: A New Approach Based on Probabilistic Feature Weighting With Positive and Negative Examples *IEEE Transactions on Image Processing*, Vol. 15, No. 4, April 2006. [10]

T. P. Minka, R. W. Picard. Interactive learning using a Society of Models. In *IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco, pp. 447–452, 1996. [11]

M. S. Lew. Next generation Web Searches for Visual Content. *IEEE Computer*. Vol 33, No 11, pp. 46–53, 2000. [12]

L. L. Thurstone. A Law of Comparative Judgement. *Psychological Review*, 34:273-286, 1927. [13]

D. M. Ennis and N. L. Johnson. Thurstone-Shepard Similarity Models as Special Cases of Moment Generating Functions. *Journal of of Mathematical Psychology*, 37:104-110, 1993. [14]

A. Tversky. Features of Similarity. *Psychological Review*, 84(4):327-352, 1977. [15]

F G. Ashby and N. A. Perrin. Toward a Unified Theory of Similarity and Recognition. *Psychological Review*, 95(1):124-150, 1998. [16]

Y. Rubner, C. Tomasi and L. J. Guibas. A Metric for Distribution with Applications to Image Databases. In *IEEE ICCV*, 1998. [17]

M.J. Swain, and B.H. Ballard. Color Indexing. *International Journal of Computer Vision*. Vol 7, No 1, 11 – 32, 1991. [18]

J. J. Rocchio Jr. Relevance Feedback in Information Retrieval. In *The Smart System – Experiments in Automatic Document Processing*. Englewood Cliffs, NJ: Prentice-Hall, pp. 313–323, 1971. [19]

E. Ide. New Experiments in Relevance Feedback. In *The Smart System-Experiments in Automatic Document Processing*. Englewood Cliffs, NJ: Prentice-Hall, pp. 337–354, 1971. [20]

G. Salton. Relevance Feedback and the Optimization of Retrieval Effectiveness. In *The Smart System – Experiments in Automatic Document Processing*. Englewood Cliffs, NJ: Prentice-Hall, pp. 324–336, 1971. [21]

Y. Ishikawa, R. Subramanya and C. Faloutsos. Mindreader: Querying Databases Through Multiple Examples. In *24th International Conference on Very Large Databases*, New York, pp. 433–438, 1998. [22]

Y. Rui, T. S. Huang and S. Mehrotra. Content-Based Image Retrieval with Relevance Feedback in MARS. In *IEEE International Conference on Image Processing*, Santa Barbara, CA, pp. 815–818, 1997. [23]

H. Müller, W. Müller, D. M. Squire, S. Marchand-Maillet and T. Pun. Strategies for Positive and Negative Relevance Feedback in *Image Retrieval. Computer Vision Group*, Computer Center, University of Geneva, Geneva, Switzerland, Technical Report, 2000. [24]

J. J. Rocchio Jr. Relevance Feedback in Information Retrieval. In *The Smart System – Experiments in Automatic Document Processing*. Englewood Cliffs, NJ: Prentice-Hall, pp. 313–323, 1971. [25]

N. Vasconcelos and A. Lippman. Learning from User Feedback in *Image Retrieval Systems*. Neur. Inf. Process. Syst., 1999. [26]

C. Meilhac and C. Nastar. Relevance Feedback and Category Search in Image Databases. In *IEEE International Conference on Multimedia Computing and Systems*, Florence, Italy, pp. 512–517, 1999. [27]

M. Crucianu, M. Ferecatu and N. Boujemaa. Relevance Feedback for Image Retrieval: a Short Survey. In *State of the Art in Audiovisual Content-Based Retrieval, Information Universal Access and Interaction, Including Datamodels and Languages*, Report of the DELOS2 European Network of Excellence, 2004. [28]

D. Tao and X. Tang. Random Sampling Based SVM for Relevance Feedback Image Retrieval. In *International Conference on Computer Vision and Pattern Recognition*, Washington, DC, 2004. [29]

S. Tong and E. Chang. Support Vector Machine Active Learning for Image Retrieval. In *ACM Multimedia Conference*, Ottawa, ON, Canada, 2001. [30]

Z. Su, H.-J. Zhang, S. Li, and S. Ma. Relevance Feedback in Content-Based Image Retrieval: Bayesian Framework, Feature Sub-spaces, and Progressive Learning. *IEEE Transactions on Image Processing*, Vol. 12, No. 8, pp. 924–937, Aug. 2003. [31]

J. Laaksonen, M. Koskela and E. Oja. PicSOM: Self-Organizing Maps for Content-based Image Retrieval. In *International Joint Conference on Neural Networks*, Washington, DC, 1999. [32]

I. J. Cox, T. P. Minka, T. V. Papathomas and P. N. Yianilos. The Bayesian Image Retrieval System, PicHunter: Theory, Implementation, and Psychophysical Experiments. *IEEE Transactions on Image Processing*, Vol. 9, No. l, pp. 20–37, Jan. 2000. [33]

S. D. MacArthur, C. E. Brodley and C.-R. Shyu. Relevance Feedback Decision Trees in Content-based Image Retrieval. In *IEEE Workshop on Content-based Access of Image and Video Libraries*, Hilton Head, SC, 2000. [34]

Y. Rui and T. S. Huang. Optimizing Learning in Image Retrieval. In *IEEE International Conference on Computer Vision and Pattern Recognition*, Hilton Head, SC, 2000. [35]

D. Tao and X. Tang. Nonparametric Discriminant Analysis in Relevance Feedback for Content-based Image Retrieval. In *International Conference on Pattern Recognition*, Cambridge, U.K., 2004. [36]

www.corel.com[37]

I. J. Cox, M. L. Miller, S. M. Omohundro and P. N. Yianilos. An Optimized Interaction Strategy for Bayesian Relevance Feedback. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 553-558, 1998. [38]

X. S. Zhou and T. S. Huang. Relevance Feedback in Image Retrieval: A Comprehensive Review. In *IEEE CVPR Workshop on Content-based Access of Image and Video Libraries* (CBAIVL), 2001[39]

K. Tieu and P. Viola. Boosting Image Retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition*, South Carolina, 2000. [40]

A.A. Goodrum, M.M. Bejune and A.C. Siochi. A State Transition Analysis of Image Search Patterns on the Web. *In International Conference on Image and Video Retrieval*, pages 269–278. Lecture Notes in Computer Science, Vol. 2728, Springer, 2003. [41]

A. Jaimes, K. Omura, T. Nagamine and K. Hirata. Memory Cues for Meeting Video Retrieval. In *CARPE Workshop*, ACM Multimedia, 2004. [42]

T. Nagamine, A. Jaimes, K. Omura and K Hirata. A Visuospatial Memory Cue System for Meeting Video Retrieval. In *ACM Multimedia* (Demonstration), 2004. [43]

Le Thi Lan. Interface de visualisation avec retour de pertinence pour la recherche d'images. In *des 1ères Rencontres Jeunes Chercheurs en Recherche d'Information*, 2004. [44]

B. Moghaddam, Q. Tian and T. Huang. Spatial Visualization for Content-Based Image Retrieval. *Technical Report* 2001-53. MITSUBISHI ELECTRIC RESEARCH LABORATORIES, February 2002. [45]

M. Nakazato and T. S. Huang. 3D MARS : Immersive Virtual Reality for Content-based Image Retrieval. *IEEE International Conference on Multimedia and Expo*, 2001. [46]

R. S. Torres, C. G. Silva, C. B. Medeiros and H. V. Rocha. Visual Structures for Image Browsing. *ACM CIKM'03*, New Orleans, Louisiana, USA, November 2003. [47]

D. Deng J. Zhang and M. Purvis. Visualisation and Comparison of Image Collections based on Self-organised Maps. *Australasian Workshop on Data Mining and Web Intelligence*, Dunedin. Conferences in Research and Practice in Information Technology, Vol. 32, 2004. [48]

G.P. Nguyen and M. Worring. Interactive Access to Large Image Collections Using Similarity-based Visualization. *Journal of Visual Languages and Computing*, 2006. [49]

S. Boutemedjet and D. Ziou. A Graphical Model for Context-Aware Visual Content Recommendation. *IEEE Transactions on Multimedia*, Vol. 10, No. 1, Jan. 2008. [50]

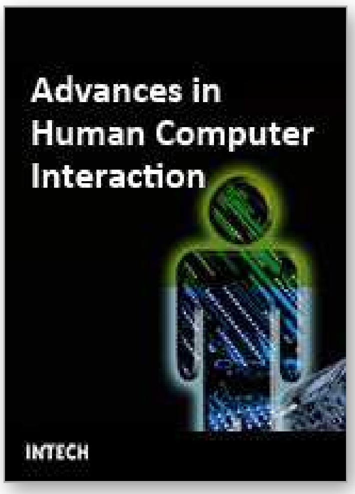
E. Bertini, A. Cali, T. Catarci, S. Gabrielli and S. Kimani. Interaction-based Adaptation for Small Screen Devices. *Lecture Notes in Computer Science* 3538, 277-281, 2005. [51]

L. Q. Chen, X. Xie, X. Fan, W. Y. Ma, H. J. Zhang and H. Q. Zhou. A Visual Attention Model for Adapting Images on Small Displays. *Multimedia Systems* Vol. 9, No. 4, 353-364, 2003. [52]

http://images.google.com/ [53]

http://www.wallpaperbase.com [54]

http://research.microsoft.com/research/downloads/Details/b94de342-60dc-45d0-830b-9f6eff91b301/Details.aspx [55]



Advances in Human Computer Interaction

Edited by Shane Pinder

ISBN 978-953-7619-15-2
Hard cover, 600 pages
Publisher InTech
Published online 01, October, 2008
Published in print edition October, 2008

In these 34 chapters, we survey the broad disciplines that loosely inhabit the study and practice of human-computer interaction. Our authors are passionate advocates of innovative applications, novel approaches, and modern advances in this exciting and developing field. It is our wish that the reader consider not only what our authors have written and the experimentation they have described, but also the examples they have set.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821