Vision Interface '99, Trois-Rivières, Canada, 19-21 May
Invited Paper

Exploratory Interfaces for Visual Information Systems

Simone Santini
Visual Computing Laboratory
University of California, San Diego
ssantini@ece.ucsd.edu

Abstract

This paper discusses image semantics and the repercussions that its correct definition has on the design of image databases. I argue that meaning can only be defined in the context of a query, and can only be revealed in the context of the whole database. Meaning is an *emergent* property that derives from the user's exploration of the image space aided by the database.

I define an interface and an interaction model that mix browsing and searching, and that allow users to explore the image space in search of meaningful images.

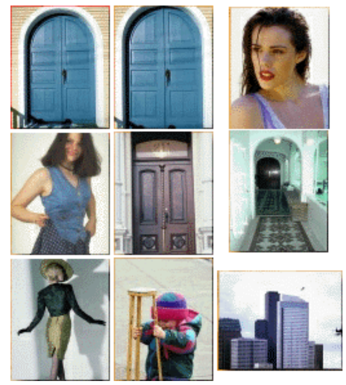

Figure 1: An example of query from a first generation image database.

1 Introduction

The Webster dictionary defines an interface as "the overlap where two theories or phenomena affect each other, or have links with each other." This definition is singularly on the mark when we are considering interfaces between users and visual information systems.

In most computer applications, the interface is a link between two semiotic systems (the user's actions and the system's functions) that share a common semantics. I will argue that this is not the case in image databases: the user and the database operate at two different semantic levels. I call the problem resulting from this mismatch tha *semantic gap*.

The semantic gap is best exemplified by making reference to Fig. 1, which is a typical example of query and answer from a current image database. The query is the image in the top-left corner, and the similarity crierion included a weighted combination of global color, local color, edges, and texture, in which the weights can be selected using four "knobs" in the interface.

Some of these images are acceptable answers to the query, while others will appear quite out of place. We can show, however, that most of them are not out of place at all. The woman face in third position, for instance, is there because the forehead of the woman is very similar in structure and position to the arch above the door in the query. The fourth image is there because the vest of the woman is similar in color and position to the door in the query, and so on. The database obviously is making some sense out of the similarity criterion that we selected. The problem is that the sense that the database is making is not what the user wanted.

We can analyze the situation as follows. The user had some *semantic specification* in mind ("I want to see images of old doors"). He found an example of an old door and concocted a similarity criterion that, according to him, captured a notion of similarity that could *induce* that semantic (the door is blue, and it has a fairly well defined structure). The database used this similarity to sort the images and returned the best results. The similarity did in fact *induce* a semantics in the database images. Alas, not the right one. This results in the *semantic gap*. The user has a fairly rich semantics in mind when he starts selecting a similarity criterion, but the tools that the database offer him are inadequate to express it.

Connected to this, there is the problem of query refinement. Receiving an answer like that of Fig. 1 is not bad at all, *providing that one knows how to change the similarity criterion in order to improve it.* In this sense, the database fails: it is not clear how one should manipulate the weights of the similarity criterion in irder to improve the answer. One doesn't know whether we should give more weight to color, less weight to texture or whatnot. In fact, our four knobs probably don't give us enough expressive power to let us express the similarity criterion we would need.

We propose that the meaning of an image is *contextual* (depends on the conditions under which the query is made) and *differential* (is apparent if the image is placed in opposition to other images which don't share the same meaning).

These ideas lead to the design of a different type of image database. In the system I am presenting, the semantics is not an intrinsic property of the images, but an *emergent* property of the interaction between the user and the database. The interface between man and machine assumes a preponderant role in this new organization.

2 Meaning

In syntactic databases, the meaning[1] of a record is compositional: it is a function of the syntactic structure of the record and of the meanings of its components.

Let $\mathcal{Q}$ be the set of all possible queries for a database. Then the meaning of a record, or a well formed fragment of a record R, can be defined as a function

$$[R] : Q \rightarrow \{\text{yes, no}\} \tag{1}$$

such that $[R](q) = \{\text{yes}\}$ if the record r satisfies the query q. Compositionality implies that, if a record is produced by a rule like

$$j : R \rightarrow \alpha_1 R_1 \alpha_2 \cdots \alpha_n R_n \alpha_{n+1} \tag{2}$$

where α_i are terminal symbols of the data definition language, R_i are non terminal symbols, and j is the label of the production rule, then the meaning of R is:

$$[R] = f_j([R_1], [R_2], \ldots, [R_n]). \tag{3}$$

The meaning of the whole record does not depend on the syntactic structure of the non terminals.

This property does not hold for images:

> The most naïve way of formulating the problem is: are there iconic sentences and phonemes? Such a formulation undoubtedly stems from a sort of verbocentric dogmatism, but in it ingenuousness it conceals a serious problem. [Eco, 197]

Eco answers negatively to the question posed in this quote. It is true that images posses certain semantic units in the form of objects, but two factors prevent us from equating images and language sentences: objects are not further decomposable using linguistic means, and they do not fully represent the meaning of images. The second property is of the utmost interest to us, since it opens up the possibility of capturing some image meaning without object recognition.

[1] In this paper, I will commit the slight imprecision of using the terms "meaning" and "semantics" interchangeably.

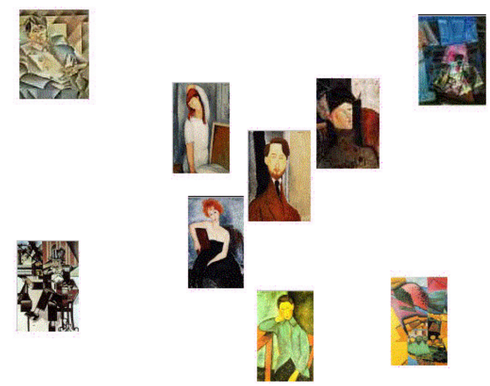

Figure 2: A Modigliani portrait placed in a context that suggests "Painting."

Semantic level beyond the objects are used very often in evocative scenarios, like art and advertising [Caliani et al., 1998]. There is, for instance, a fairly complex theory of the semantics associated with color [Itten, 1961], and with certain artistic representational conventions [Gombrich, 1965].

The full meaning of an image depends not only on the image data, but on a complex of cultural and social conventions in use at the time and location of the query, as well as on other contingencies of the context in which the user interacts with the database. This leads us to reject the somewhat Aristotelean view that the meaning of an image is an immanent property of the image data. Rather, the meaning is *created* during a subjective process of interpretation of the image. A query process does not filter images based on an illusory pre-existing meaning but creates meaning through the interaction of the user and the images.

This interpretation process is not unique for a given image and situation, but depends on the context in which the images are presented. Consider the images of Fig. 2. The image at the center is a Modigliani portrait. The same image is placed on a different context in Fig. 3. When subjects were asked to define the central image using a few words, the word "painting" was more frequently used to describe the image in the context of Fig. 2, and the word "face" was more frequently used to describe it in the context of Fig. 3.

All these somewhat overly philosophical observations point in the same direction: interactivity (and, consequently, interfaces) is an essential component of visual information systems. The idea of querying an image database based on content is illusory, and not (or not only) for mere technological immaturity. The characteristics of image meaning make the interaction with the user a necessity. An important re-

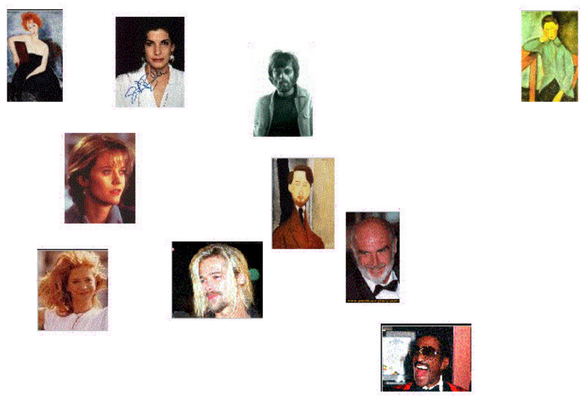

Figure 3: A Modigliani portrait placed in a context that suggests "Face."

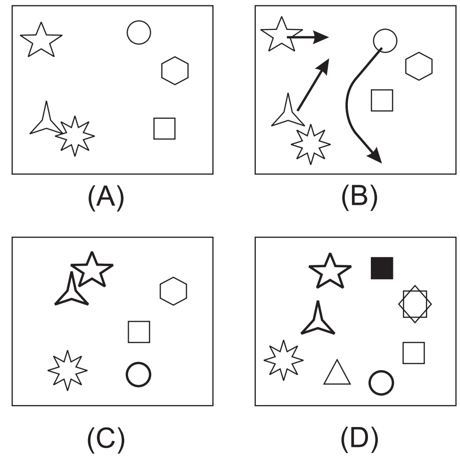

Figure 4: Schematic description of an interaction using an exploratory interface.

sult of this constant user supervision is the blurring of the distinction between query and browsing. The user is browsing an image space when trying to understand the current similarity criterion used by the database, and is querying the database when asking it to reorganize the image space to avoid certain undesired association. I call this union of browing and querying *exploration*. In the following section we will see how exploration is embodied in a visual database interface.

3 Emergent Semantic Interface

The observations in the previous sections give us indications on how and interface should work. We can summarize the conclusions in the following two points:

- Effective communication from the database to the user can take place only if the user is aware of the context in which the database is oprtating. Showing images is not sufficient: it is necessary to show the relations between images.

- Direct intervention on the parameters of the distance measure, as postulated in today's interface is ineffective because it takes place at the wrong semantic level. The user should only manipulate quantities of immediate semantic significance. In the context of an interface like as the one we are outlining, the relation between images provide such quantities.

We can add a more practical observation to the second point. The necessity to control the parameters of a distance measure limits severely the flexibility and adaptability of the measure. Many similarity measures of interest can contain up to a hundred parameters [Santini and Jain, 1999a]. It is obviously impossible to present the user an interface with a hundred knobs and this forcibly reduces the number of parameters that can be adapted.

The two principles above are encoded in what I call an *exploratory* interface. The context in which the database is placing the images is made manifest by displaying a map showing an approximation of the relative positions of the images in the feature space according to the current similarity criterion. We can consider this as a sort of cognitive map of the database in its current situation. The user can browse the map and, at the same time, relocate some of its images to better reflect her own similarity concept (her own cognitive map for the current query). The database responds by uptating the similarity criterion it is using and by updating the display.

An user interaction using an exploratory interface is shown schematically in Fig. 4. In Fig. 4.A the database proposes a certain distribution of images (represented as shapes) to the user. The distribution of the images reflects the similarity interpretation given by the database. For instance, the triangular star is considered very similar to the octagonal star, and the circle is considered similar to the hexagon. In Fig. 4.B the user moves some images around to reflect his own interpretation of the relevant similarities. The result is shown in Fig. 4.C. According to the user, the pentagonal and the triangular stars are are quite similar to each other, and the circle is quite different from both of them.

As a result of the user assessment, the database will create a new similarity measure, and re-order the images, yielding the configuration of Fig. 4.D. The pentagonal and the triangular stars are in this case considered quite similar (although they were moved from their intended position), and the circle quite different. Note that the result is not a simple

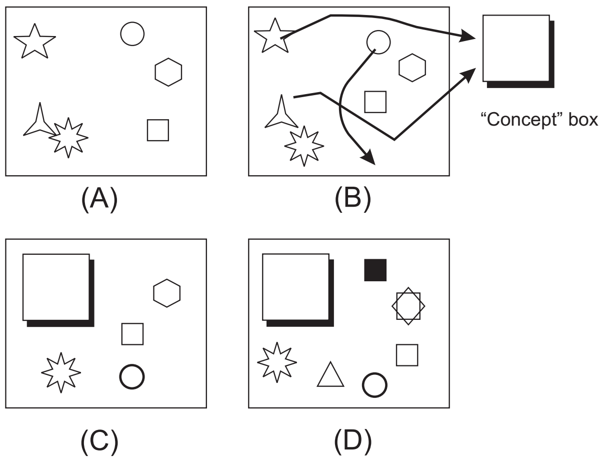

Figure 5: Interaction involving the creation of concepts.

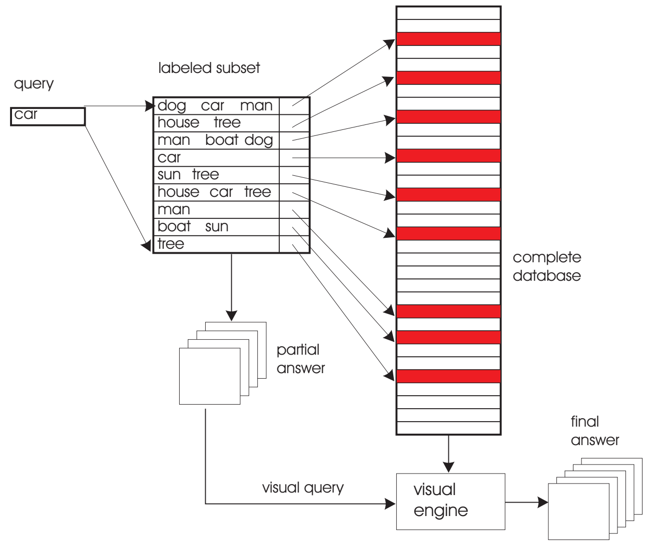

Figure 6: Schematic representation of a search involving a visual dictionary.

rearrangement of the images in the interface. For practical reasons, an interface can’t present more than a small fraction of the images in the database. Typically, only the 100-300 images most relevant to the query are displayed. The reorganization consequent the user interaction involves the whole database. Some images will disappear from the display (the hexagon in Fig. 4.A), and some will appear (e.g. the black square in Fig. 4.D).

In addition to this basic feedback mechanism, our interface provides tools for browsing the display (presented in section 4), and a number of auxiliary operators. I will briefly describe two of them: *visual concepts* and the *visual dictionary*.

3.1 Visual Concepts

The interface allows the definition and placement of visual concepts. A visual concept is simply a set of images that, for the purpose of the current application, can be considered as equivalent or almost equivalent and can be assigned a defined semantic identity. Images forming a visual concept can be dragged into a “concept box” and, if necessary, associated with some text (the text can be used to retrieve the concept and the images similar to it). The visual concept can be then transformed into an icon and placed on the screen like every other image.

Fig. 5 is an example of interaction involving the creation of visual concepts. Fig. 5.A contains the answer of the database to a user query. The user considers the pentagonal and the triangular starts as two instances of a well defined linguistic concept, and opens a *concept box*, dragging the two stars inside it. The box is then used as an icon to replace the images in the display space.

From the point of view of the interface, a concept is a group of images that occupy the same position in the display space. In addition, it is possible to attach meta-data information (typically textual) to a concept, and use it to retrieve the concept and place it in the interface.

3.2 Visual Dictionary

The second tool (the visual dictionary) derives from the same need to integrate textual information in visual databases. It is well known that attaching labels to a database suffers from two drawbacks: it is an extremely expensive operation (which limits its applicability to applications with high added value), and usually doesn’t capture all the semantics of an image. In a visual dictionary, we label a subset of a database and use the results of the textual search as a starting point for the visual query.

The structure of a visual dictionary is shown in Fig. 6. Let us assume that a user is looking for some fairly romantic images of old cars in quiet country roads. We have a large database D of images, and a subset $A \subset D$ that has been labeled (or for which every image has some text attached). Note that A might not contain the images that we are looking for, and that its labeling might be too coarse for the semantics that we are considering (e.g. images might not be labeled according to their romantic content, or of whether they are city or country images.) On the other hand, A will probably contain some examples of cars, and we will be able to retrieve them as a (partial) match to the query “old cars in country roads” using standard information retrieval techniques [Riloff and Hollaar, 1996]. Although these cars are not what we are looking for, we can use them as visual examples to start a visual query in the whole database D.

The visual dictionary solves two major problems of text and visual databases:

- It overcomes the problems of labeling schemes. It is not necessary to label the whole database, or to try to capture in the next all the minutiae of an image. We don't expect a good answer from the visual dictionary, but just enough examples to start a visual query.

- The visual dictionary provides a convenient way to start a visual search. Apart from the idea of sketching what one is looking for, there is no comonly accepted way of posing a query to a database. A visual dictionary is a tool for starting a query.

In addition to all this, the visual dictionary is easily integrated with visual concepts. A visual concept can be seen as a dictionary entry generated by the user in the context of a query.

4 Exploration Interface Operators

The exploratory interface is defined in terms of a number of operators between three spaces [Gupta et al., 1997]: the *feature space*, the *query space*, and the *display space*.

The *Feature space* $\mathcal{F}$ is the space of the coefficients of a suitable representation of the image. The feature space is topological, but not metric. The feature space is independent of the query and is defined when the database is created. I will not consider operators in the feature space, but some are defined in [Santini and Jain, 1999b].

When the feature space is endowed with a metric, the result is the *query space* $\mathcal{Q}$. The metric of the query space is derived from the user query, so that the distance from the origin of the space to any image defines the "dissimilarity" of that image from the current query. The *Display space* $\mathcal{D}$ is a low dimensional space (0 to 3 dimensions) which is displayed to the user and with which the user interacts. The distribution of images in the display space is derived from that of the query space. In this paper I will consider only two-dimensional display spaces (as implemented in a window on a computer screen.) For the sake of convenience, I also assume that every image in the visualization space has attached a number of *labels* λ_i drawn from a finite set. Examples of labels are the visual concepts to which an image belongs. The conventional label α is assigned to those images that have been selected by the user and placed at a given position on the screen.

4.1 Operators in the Query Space

The feature space, endowed with a similarity measure derived from a query, becomes the query space. The "score" of an image is determined by its distance from the origin. The determination of the geometry of the query space is in general quite complicated, and is beyond the scope of this paper. I will just assume that every image is represented as a set of n number (which may or may not identify a vector in an n-dimensional vector space) and that the query space is endowed with a distance function that depends on m parameters.

The feature sets corresponding to images x and y are represented by x^i and y^i, $i = 1, \dots n$, and the parameters by ξ^μ, $\mu = 1, \dots, m$. Also, to indicate a particular image in the database I will use either different Latin letters, as in x^i, y^i or an uppercase Latin index. So x_I is the I-th image in the database ($1 \leq I \leq N$), and x_I^j is the corresponding feature vector.

The parameters ξ^μ are a representation of the query, and are the values that determine the distance function. Given the parameters ξ^μ, the distance function in the query space is $f(x^i, y^i; \xi^\mu)$. Depending on the situation, I will write $f_\xi(x^i, y^i)$ in lieu of $f(x^i, y^i; \xi^\mu)$.

As stated in the previous section, the feature space is topological but not metric. Rather, its intrinsic properties are characterized by the functional $L : \mathbb{R}^m \to L^2(\mathbb{R}^n \times \mathbb{R}^n, \mathbb{R}^+)$ which associates to each query ξ^μ a distance function $L(\xi^\mu) = f_{\xi^\mu}$.

A query q, characterized by a vector of parameters ξ^μ, can also be seen as an operator q_ξ which transforms the feature space into the query space. If L is the characteristic functional of the feature space, then $q_\xi L = L(\xi)$ is the metric of the query space.

Once the feature space $\mathcal{F}$ space has been transformed into the metric query space Q, other operations are possible [Gupta et al., 1997].

Given a feature set x^i, the *distance operator* return its distance from the query:

$$D(x^i) = f(0, x^i; \xi^\mu) \tag{4}$$

The *select by distance* operator returns all feature sets that are closer to the query than a given distance:

$$S(d) = \left\{ x^i : D(x^i) \leq d \right\} \tag{5}$$

The k-*Nearest Neighbors* returns the k images closest to the query

$$N(k)(\mathcal{F}) = \left\{ x^i : \left| \left\{ y^i : D(y^i) < D(x^i) \right\} \right| < k \right\} \tag{6}$$

It is necessary to stress again that these operations are not defined in the feature space $\mathcal{F}$ since that space is not endowed with a metric. Only when a query is defined does a metric exist.

4.2 The Display Space

The display operator ϕ projects image x^i on the screen position X^Ψ, $\Psi = 1, 2$ in such a way that

$$d(X^\Psi, Y^\Psi) \approx f(x^i, y^i; \xi^\mu) \tag{7}$$

I use a simple elastic model to determine the position of images in the display space. First, the database is interrogated to determine the P images closer to the query. The display space will be concerned only with these images. In general, I use $100 \leq P \leq 300$. A few hundred images are in general sufficient to give the user a fair idea of the image distribution in the database, and don't clutter the display with irrelevant information. Let $f(x_I^k, x_J^k; \xi^\mu)$, $I, J \leq P$ be the distance between the I-th and the J-th image in the database, with $0 \leq f(x_I^k, x_J^k; \xi^\mu) \leq 1$. Also, let X_I^Ψ be the coordinates of the I-th image in the display space, and $d(X_I^\Psi, X_J^\Psi)$ the Euclidean distance between images I and J in the display space. In a given configuration $\{X_I^\Psi, i = 1, \ldots, P\}$ the images are placed at interface coordinates X_I that minimize the functional

$$E = \sum_{i,j=1}^{Q} \left(d(X_I^\Psi, X_J^\Psi) - f(x_I^k, x_J^k; \xi^\mu) \right)^2 \tag{8}$$

Standard techniques can solve this optimization problem and find the optimal configuration of the display space. The result is an operator that we write:

$$\phi(x_I^k; f_\xi) = (X_I^\Psi, \emptyset). \tag{9}$$

The parameter f_ξ reminds us that the projection that we see on the screen depends on the distribution of images in the query space which, in turn, depends on the query parameters ξ^μ. The notation $(X_I^\Psi, \emptyset)$ means that the image x_I is placed at the coordinates x_I^Ψ in the display space, and that there are no labels attached to it (that is, the image is not anchored at any particular location of the screen, and does not belong to any particular visual concept).

A configuration of the display space is obtained by composing the projection operator with the k-nearest neighbor operator

$$\phi(N(k)(\mathcal{Q})) = \phi(N(P)(\mathcal{F}); f_\xi) \tag{10}$$

where $\aleph_I$ is the set of labels associated to the I-th images. This operator displays the k images closest to the query in their appropriate configuration.

The Place Operator The place operator moves an image from one position of the display space to another, and attaches a label α to the images to "glue" it to its new position. The operator that places the I-th image in the display is $\zeta_I : \mathcal{Q} \rightarrow \mathcal{Q}$ with:

$$\begin{aligned} \zeta_I \left\{ (X_J^\Psi, \aleph_J) \right\} &= \left(\left\{ (X_J^\Psi, \aleph_J) \right\} - \left\{ (X_I^\Psi, \aleph_I) \right\} \right) \\ &\quad \cup \left\{ (\tilde{X}_I^\Psi, \aleph_I \cap \alpha) \right\} \end{aligned} \tag{11}$$

where $\tilde{X}$ is the position given to the image by the user.

Other operators for the manipulation of the display space include navigation operations like panning and zooming.

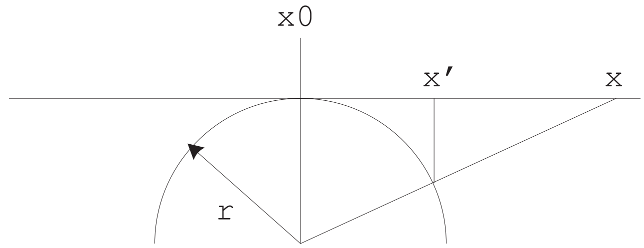

Figure 7: One-dimensional version of the fish eye lens operator.

More sophisticated operations are possible, such as the "fish eye" projection []. The fish-eye operator represents the whole cognitive map in a limited screen space by assigning a large portion of the screen to the portion of the map that the user is examining, and compressing remote portions of the cognitive map into a smaller screen area. A one-dimensional example of fish eye is shown in Fig. 7 The horizontal like represents the display, and x_0 is the position of the display space which the user is examining and on which he has centered the fish-eye lens. The point x in the undeformed display space is projected into the screen point

$$x' = x_0 + \frac{r}{\sqrt{x^2 + r^2}}(x - x_0) \tag{12}$$

where r is the radius of the lens.

Finally, formal operators can be defined to place visual concepts and the results from the visual dictionary in the display. These operators were described in [Santini and Jain, 1999b] and will not be described here.

4.3 Query Creation

When the user moves images around the interface, he imposes a certain number of constraints of the form $d(X_I, X_J) = d_{xy}$. Assume that the user takes a set T of images and places them in certain positions of the interface, so that, for all pairs $(X_I, X_J) \in T \times T$, the value d_{xy} is given. The query can then be determined by solving the system of equations:

$$f(x^i, y^i; \xi^\mu) = d_{XY} \quad X, Y \in T \tag{13}$$

in the unknown ξ^μ. This system of equations is in general underconstrained, since the user will typically move around up to a dozen images, and the distance measures may have a hundred parameters ξ^μ. We identify a solution using the concept of *natural distance*. For each feature space upon which the database operates, it is possible to define a natural distance which, ex hypothesis, makes the curvature of the space zero. Intuitively, a natural distance is the feature space equivalent of an isotropic and uniform distance. Let ξ_0 be

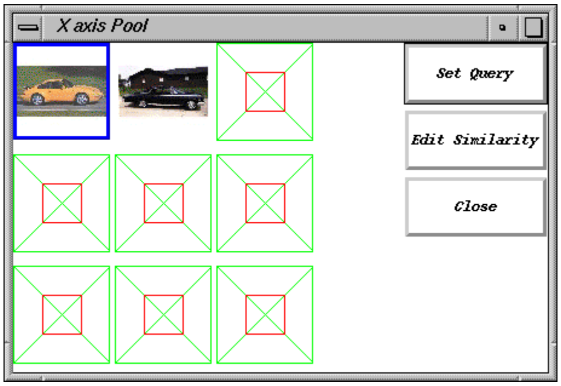

Figure 8: The initial concept of car that we use.

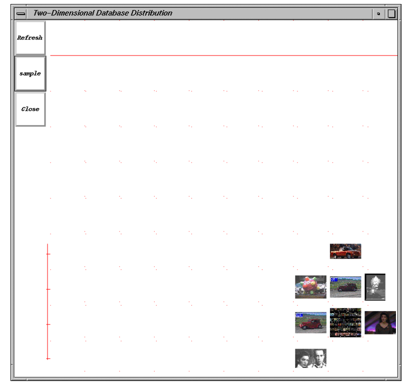

Figure 9: The result of a query with our first "car" concept.

the parameter vector corresponding to the natural distance, and $C(\xi, \xi_0)$ the curvature between the metric given by ξ_0 and that given by ξ. The query formation operator takes the distances d_{XY} imposed by the user and modifies the metric of the query space by finding the ξ which minimizes

$$\sum_{IJ} \left(f(x_I^i, x_J^i; \xi) - d_{X_I, X_J} \right)^2 + C(\xi, \xi_0) \qquad (14)$$

In other words, we select a metric for the query space that compromises between the satisfaction of the constraints imposed by the user and the minimization of the curvature with respect to the natural distance.

The creation of a query can be seen as an operator

$$\chi : \mathcal{D} \to \mathbb{R}^m : \left\{ (X_I^\Psi, \aleph_I) \right\} \mapsto \xi^\mu. \qquad (15)$$

When the new parameters ξ are available, the system reorganizes the database according to the new metric and creates a new configuration of the k images closest to the query to be shown to the user. The details of the creation of a similarity measure depend on the characteristics if the search engines. Details are reported in [Santini, 1998, Santini and Jain, 1997].

5 The Interface at Work

This section presents an example of interface based on the principles illustrated in the previous section, and implemented in the database system El Niño[Santini and Jain, 1999a]. To give an example of a typical interaction session with El Niño, consider a query in which we are looking for some old cars. At the beginning our ideas are quite fuzzy, and we set to explore the database. We have a few cars in our "labeled" subset of the database, and we start defining the concept of car as in Fig. 8. The result of a query using this concept is shown in Fig 9 This answer is not satisfactory, but it contains the seeds from

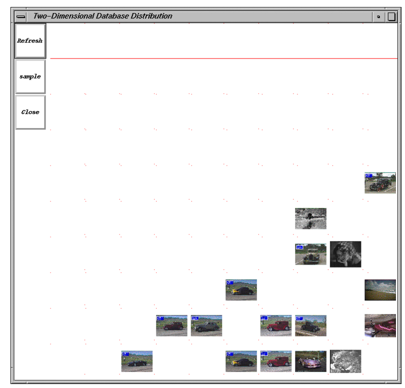

Figure 10: The result of a query with our second "car" concept.

which we can proceed towards more interesting areas of the image space. We select a few of the cars in the display and add them to the concept of car. We go through the stage of Fig. 10 until, at the end of our query, our concept of "car" has become that of Fig. 11. During this interaction, our idea of what would be an answer to the query changed continuously as we learned what the database had to offer, and redefined our goals based on what we saw.

6 Conclusions

In this paper I have defined a new mode of interaction for image databases which I called *exploration*. The motivation for the introduction of this model comes from an analysis of the semantics of images in the context of an image database. In traditional databases, the meaning of a record is a function from the set of queries to a set of truth values. The meaning of an image, on the other hand, can only be revealed by the comparison of an image with other images in the feature space.

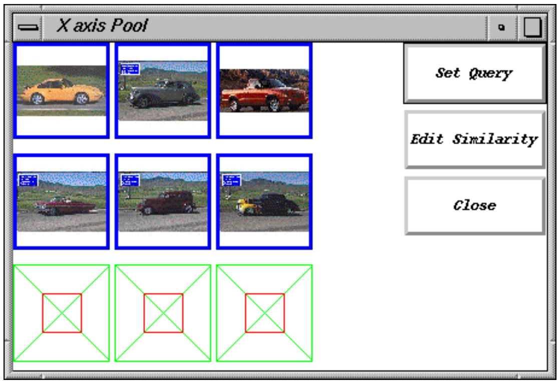

Figure 11: The final concept of car after the interaction.

These observations led us to define a new paradigm for database interfaces in which the role of the user is not just asking queries and receiving answers, but a more active exploration of the image space. The meaning of an image is *emergent*, in the sense that it is a product of the dual activities of the user and the database mediated by the interface.

I have proposed a model of interface for active exploration of image spaces. In this interface, the role of the database is to *focus* the attention of the user on certain relations that, given the current database interpretation of image meanings, are relevant.

It is interesting to note that the distinction between the role of the user and the role of the database is blurred in this model. In very general terms, the role of the database is to *focus* the attention of the user on certain relations that, given the current interpretation of meaning in the database, are deemed relevant. The database does this by displaying a subset of relevant images and their relations in the similarity criterion that is used to define the meaning.

The role of the user is exactly the same. By displacing images in the interface plane, the user focuses the attention on the database on certain relation between images that, given the user interpretation of the meaning of the displayed images, are relevant. Both systems, the database and the user, will adjust their similarity measure based on the response of the other system. The fact that we expect more flexibility from the database rather than from the user (i.e. the database should adapt its similarity measure, while the user has a relatively stable idea of what he/she wants) makes the difference between the two a matter of degree rather than a categorical distinction.

Future plans for El Niño include the design of perceptually more comprehensive interface, in which aural and haptic clues play are used to supplement visual clues.

References

[Caliani et al., 1998] Caliani, M., Pala, P., and Del Bimbo, A. (1998). Computer analysis of TV spots: The semiotics perspective. In *Proceedings of the IEEE International Conference on Multimedia Computing and Systems, Austin, TX*.

[Eco, 197] Eco, U. (197). *A Theory of Semiotics*. Indiana University Press, Bloomington.

[Gombrich, 1965] Gombrich, E. H. (1965). *Art and Illusion. A study in the pshychology of pictorial representation*. Pantheon Books.

[Gupta et al., 1997] Gupta, A., Santini, S., and Jain, R. (1997). In search of information in visual media. *Communications of the ACM*, 40(12):35–42.

[Itten, 1961] Itten, J. (1961). *The art of Color*. Reinhold Pub. Corp., New York.

[Riloff and Hollaar, 1996] Riloff, E. and Hollaar, L. (1996). Text databases and information retrieval. *ACM Computing Surveys*, 28(1):133–135.

[Santini, 1998] Santini, S. (1998). *Explorations in Image Databases*. PhD thesis, Univerity of California, San Diego.

[Santini and Jain, 1997] Santini, S. and Jain, R. (1997). Similarity is a geometer. *Multimedia Tools and Applications*, 5(3). available at http://www-cse.ucsd.edu/users/ssantini.

[Santini and Jain, 1999a] Santini, S. and Jain, R. (1999a). Integrated browsing and querying of image databases. *IEEE Multimedia Magazine*. (submitted).

[Santini and Jain, 1999b] Santini, S. and Jain, R. (1999b). User interfaces for emergent semantics in image databases. In *Proceedings of the 8th IFIP Working Conference on Database Semantics (DS-8), Rotorua (New Zealand)*.