Visual Similarity Measurement with the Feature Contrast Model

Horst Eidenberger*, Christian Breiteneder
Vienna University of Technology, Institute of Software Technology and Interactive Systems,
Favoritenstrasse 9-11, 1040 Vienna, Austria

ABSTRACT

The focus of this paper is on similarity modeling. In the first part we revisit underlying concepts of similarity modeling and sketch the currently most used VIR similarity model (Linear Weighted Merging, LWM). Motivated by its drawbacks we introduce a new general similarity model called Logical Retrieval (LR) that offers more flexibility than LWM. In the second part we integrate the Feature Contrast Model (FCM) in this environment, developed by psychologists to explain human peculiarities in similarity perception. FCM is integrated as a general method for distance measurement. The results show that FCM performs (in the LR context) better than metric-based distance measurement. Euclidean distance is used for comparison because it is used in many VIR systems and is based on the questionable metric axioms. FCM minimizes the number of clusters in distance space. Therefore it is the ideal distance measure for LR. FCM allows a number of different parameterizations. The tests reveal that in average a symmetric, non-subtractive configuration that emphasizes common properties of visual objects performs best. Its major drawback in comparison to Euclidean distance is its worse performance (in terms of query execution time).

Keywords: Visual Information Retrieval, Content-based Image Retrieval, Content-based Video Retrieval, Visual Similarity Measurement, Similarity Modeling, Linear Weighted Merging, Logical Retrieval, Feature Contrast Model, Boolean Retrieval, Vector Space Model

1 INTRODUCTION

Content-based Image Retrieval (CBIR) and Content-based Video Retrieval (CBVR) are two research directions of Multimedia (or Media Processing) systems that have been very active in the last couple of years. There is a trend to unify concepts and methods from CBIR and CBVR under one umbrella and the MPEG-7 standard is a first step in this direction. In this paper we will refer to both CBIR and CBVR as Visual Information Retrieval[4] (VIR). Essentially, VIR research has undergone three phases with different focus: (1) feature design and indexing methods, (2) user-interfaces and iterative refinement and (3) benchmarking (now). To overcome the big open problems of VIR[19] we think it being necessary to emphasize careful feature and similarity modeling.

The focus of this paper is on similarity modeling, which is probably the most neglected area of VIR research. First we sketch the currently most used VIR similarity model (Linear Weighted Merging, LWM) and point out its major weaknesses. Then we describe a new general similarity model called Logical Retrieval (LR) that offers more flexibility than LWM without suffering from its drawbacks. In the third step we integrate the Feature Contrast Model (FCM), developed by Tversky[20] into this environment. The FCM is based on psychological observations of human similarity perception. We integrate it in LR as a general method for feature-based distance measurement.

Utilizing FCM for VIR is not a new idea: it was first performed by Santini and Jain[15, 16], who built a unified similarity theory integrating geometric and set theoretic approaches. The FCM defines a family of distance measures that are very attractive for VIR: It is possible to define asymmetric similarity (e.g. "how similar is image A to B?" instead of "how similar are A and B?") and the model is generally less restrictive than distance functions based on the metric axioms (e.g. Minkowski distances).

The following Section 2 offers background information on similarity structures and distance measurement. Section 3 reviews similarity modeling in VIR, the LWM approach and describes LR as a more flexible model than LWM. Section 4 integrates the FCM in LR and investigates whether the integrated model still preserves the idea of Tversky's FCM. In Section 5 the FCM is implemented in a prototype as a general-purpose distance measure and tested on image data. Performance results are analyzed in comparison to Euclidean distance as a standard VIR distance measure.

* eidenberger@ims.tuwien.ac.at; phone 43 1 58801-18853; fax 43 1 58801-18898; www.ims.tuwien.ac.at

2 BACKGROUND

Subsequently, we define the term similarity structure as the fundamental concept for distance respective similarity measurement. Subsection 2.2 sketches distance measurement based on the metric axioms as it is usually used in VIR systems. Finally, Subsection 2.3 briefly describes alternative axiomatic systems for similarity structures including the FCM.

2.1 Similarity structures

Let E be an arbitrary set of objects. According to Sint[17] a similarity structure (or: a similarity measure) for the elements e_i of E is defined as a relation respective function over a set of pairs ExE of objects (represented as numerical feature vectors). The given measurements have to be somehow transformed into this relation. The list of possible similarity structures S over ExE includes[17]:

- S_1: S is an Euclidean distance over ExE. This measure assumes that feature space has Euclidean geometry (fulfills the metric axioms, see below).
- S_2: S is a metric over ExE. This measure makes no assumption on the geometric shape of feature space. S_2 is a generalization of S_1.
- S_3: S is symmetric and rational over ExE.
- S_4: S is a total or partial order of E.

These four are the most common similarity structures but of course, many more do exist. This definition spans an umbrella over a wide range of similarity understandings (visual, mathematical, psychological, etc.). In this paper we will investigate another method: the generation of a dichotomy of similar and not similar objects with dynamic borders over E.

2.2 Distance measurement based on the metric axioms

Usually, VIR similarity measurement follows the vector space model from information retrieval theory (e.g. in LWM, see Section 3.1). It is done by measuring the distances of feature vectors with distance functions and interpreting similarity as a point in an n-dimensional distance space. The vector space model is an applied similarity structure of type S_2. That means, it strongly relies on metric-based distance measurement. For distance measurement in (feature) vector spaces a certain type of geometry has to be considered. In VIR the feature space is usually considered to be of Euclidean shape. That means, distance measures $d()$ fulfil four conditions (metric axioms)[16]:

1. Constancy of self-similarity:

$$d(f_A, f_A) = d(f_B, f_B) \tag{1}$$

for the feature vectors f_A and f_B of two stimuli A and B (in VIR: media objects). Psychological experiments have show that self-similarity is not always the case for human similarity perception[16].

2. Minimality:

$$d(f_A, f_B) \geq d(f_A, f_A) \tag{2}$$

3. Symmetry:

$$d(f_A, f_B) = d(f_B, f_A) \tag{3}$$

Like for the constancy of self-similarity, psychological experiments have turned out that humans do not always have a symmetric similarity perception.

4. Triangle inequality:

$$d(f_A, f_B) + d(f_B, f_C) \geq d(f_A, f_c) \tag{4}$$

Distance measures that fulfil the metric axioms are Minkowski distances, the Euclidean distance and the city block measure[16]. Experimental investigations during the last fifty years have turned out that metric axioms may be too restrictive for human similarity perception. The triangle inequality (in CBIR sometimes used for query acceleration) was even falsified[16, 20]. Newer theories as the ones sketched in the next subsection suggest a better representation of human similarity perception.

2.3 Alternatives for the metric axioms

According to Santini and Jain, Monotone Proximity Structures (MPS, a system of three distance axioms) could be used to replace the metric axioms with a less rigid system[16]. As can be easily shown, MPS suffers from severe inconsistencies. One of the axioms is the dominance axiom:

$$d(x_1 y_1, x_2 y_2) > \max\{d(x_1 y_1, x_1 y_2), d(x_1 y_1, x_2 y_1)\} \tag{5}$$

Here, two stimuli *A* and *B* are compared by distance function *d()* where *A* and *B* are represented by two-dimensional feature vectors (x_1,y_1) and (x_2,y_2). For example, if the two features are the following predicates: (1) "*X* is a color image" and (2) "*X* has landscape spatial layout" (where *X* is an arbitrary stimulus) then a greyscale landscape media object can be represented by $x_1=0$, $y_1=1$ and a color landscape media object can be represented by $x_2=1$ and $y_2=1$. For this example, the dominance axiom has to be written as:

$$d((0,1),(1,1)) > \max\{d((0,1),(1,1)), d((0,1),(1,1))\} \equiv d((0,1),(1,1)) > d((0,1),(1,1)) \tag{6}$$

Obviously, no distance function *d()* exists for which equation 6 holds.

In comparison to the metric axioms and MPS, FCM is not a geometric but set-theoretic approach[20]. Basically, the idea is measuring the similarity of two stimuli (represented by feature vectors *X* and *Y*) with the formula in equation 7 (*f()* is a monotone increasing function and the non-negative parameters α, β determine, whether *s()* is symmetric ($\alpha=\beta$) or asymmetric (else) and subtractive ($\alpha>0$ or $\beta>0$) or non-subtractive (else)).

$$s(X,Y) = f(X \cap Y) - \alpha\, f(X - Y) - \beta\, f(Y - X) \tag{7}$$

The FCM is very successful in representing the properties of human similarity measurement because it allows to distinguish between symmetric and asymmetric similarity perception and accounts for non-constant self-similarity. On the other hand it does not allow measurement with constant self-similarity and can only be applied to qualitative feature vectors (predicates). The latter is because of the logical operators used in *f()*.

To overcome the second drawback, Santini and Jain developed the Fuzzy FCM (FFCM) where the numerical elements of feature vectors are transformed to truth values and the logical operators (intersection and subtraction) are replaced by fuzzy equivalents[15, 16]. In addition, they developed a geometric equivalent for FFCM by replacing the fuzzy set operators by continuous functions. This formula somehow integrates geometric and set-theoretic similarity approaches. Santini and Jain[16] present a solution for the problematic fact that in FFCM feature vector elements are considered to be independent, which is not the case in reality. Unfortunately, this approach has not been integrated with the continuous FFCM formula. Additionally, these approaches suffer from the drawback, that the major degree of freedom (the selection of the formula *f()*) had to be abandoned in favour of unification. *f()* is always the fuzzy cardinality of the given truth values.

Maybe because of these problems, it seems that using FCM for VIR is not further investigated. For example, in Smeulders et al[18] it is not mentioned any more. We think this being regrettable, because the ideas of FCM are valuable for VIR distance measurement. In Section 4 we describe a new approach to incorporate FCM in a process-oriented environment for similarity measurement (LR). Next, we design a general model for the representation of human

similarity perception.

3 SIMILARITY MODELING

In philosophy, similarity defines the relation between an object and its representation (Plato's 'image'). In VIR the philosophical similarity problem is usually subsumed under the term sensory gap[18]. Essentially, this term describes the loss of information in the (repeated) photographing process. This problem is accepted and not treated in VIR. Computer vision people are plowing this field and often, their plow is a 3D model (e.g. an active contour based on a deformable template, etc.).

Quite differently, 'similarity' in VIR is the relation of two images (stimuli). These may, but need not be representations of two objects (scenes, etc.). Thus, the VIR similarity problem is modeling the *real* similarity perception, humans have developed since they are able to use sticks for drawing in the dust. This problem has been investigated mostly by psychologists (perception theory, Gestalt theory). Researchers in other areas of work use the term 'similarity' as well but – as pointed out above – mean something different than (human perception of) visual similarity (e.g. mechanics: similarity of machines), have a more strict similarity concept (e.g. mathematics: similarity of triangles) or investigate the similarity of abstract representations of objects or images (e.g. physics: similarity in thermo-dynamics, medicine: homeopathy theory). Somehow logical, the most similar meaning of 'similarity' is used in biology for categorization of species. In this section we examine the standard VIR similarity model and develop a new one (Subsection 3.2) that is more suitable for human similarity perception.

3.1 Standard VIR similarity model

The usual approach for VIR similarity measurement is called Linear Weighted Merging (LWM) and has the following form[18] (generalized):

$$s_A(F,E) = \frac{\sum_{i=1}^{|F|}\left(w_i \sum_{j=1}^{|E|} u_{i,j} h_i\left(d_i\left(f_{i,A}, f_{i,j}\right)\right)\right)}{\sum_{i=1}^{|F|} w_i \sum_{j=1}^{|E|} u_{i,j}} \tag{8}$$

$s_A(F,E)$ is the average dissimilarity of stimulus A related to the used set of features F and the query examples in set E. The w_i are the weights for the features, $u_{i,j}$ is a binary matrix of size $|F|x|E|$ that contains a '1' for each combination of feature and query example that is used in the query. $h_i()$ can be any linear or exponential transformation of the distance values $d_i(f_{i,A}, f_{i,j})$ for feature i, stimulus A and query example j. Typical (accepted) transformations are identity and negative exponential transformation[18]:

$$h_i(d) = e^{-d} \tag{9}$$

This numerator is standardized by the denominator: the number of dimensions of distance space. Because the denominator represents a linear transformation it is often omitted. In this case we do not call $s_A()$ a similarity but a position value, because the $s_A()$ for all objects A are a partial order over the given object collection. Like the distance functions, $s_A()$ is a similarity structure (in this case of type S_4, see Subsection 2.1). Usually, the most similar objects have the smallest position values. If the second transformation (equation 9) is used, the most similar stimuli have the highest position values. The linear weighted merging formula implies that all distance measures are standardized to the same interval (usually [0,1]). Ideally, they should have the same distribution as well. The weights are usually provided by the user and usually sum up to 1.

The LWM formula does not measure the distance of an object to the origin of distance space! It is just a linear combination of distance values. One argument against this formula is that most features are (of course) not linearly related. The fundamental law of Gestalt Theory is a generalization of this fact: the whole is more than its elements. In the formula above, the maximum of derivable information is always the sum of the elements.

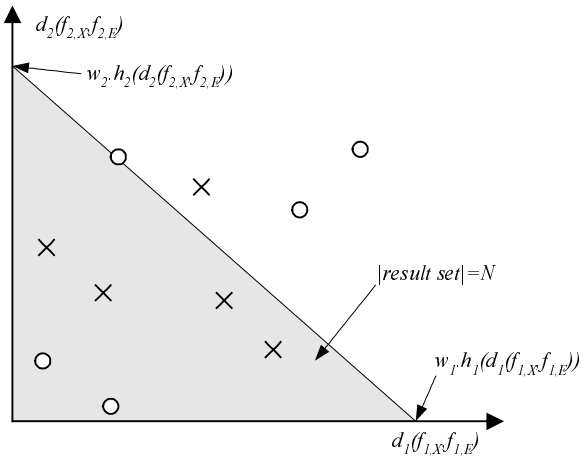

Figure 1: Similarity definition in distance space by LWM. The selected subspace is always of triangular shape. Its size depends the result set size and the elements in the queried collection. The slope of the border is determined by the weights. Thus it is impossible to sort out all irrelevant elements in iterative refinement.

One could argue, that for testing it is sufficient to know the ground truth for the query examples and use this formula to find a lower bound for the quality of a retrieval method. The ground truth according to one element is the number of similar elements in a collection. For obvious reasons this is not true. LWM just generates a partial order over all elements of a collection. To be usable for retrieval the user has to specify a number *N* for the number of similar elements in the result set. Knowing the ground truth *G* (for the given examples) per se influences the selection of *N*. If *G* is selected greater than *N* the recall improves and if *G* equals *N* the precision improves. Additionally, it is not clear, how the ground truth can be found for tests, where using multiple example objects is necessary. Finally, knowing the ground truth and using LWM is not enough to judge the quality of a method (e.g. for feature extraction). In this case it would be necessary to know the distribution of objects in distance space as well.

The next argument follows a similar direction. Usually, iterative refinement by relevance feedback is used to reduce the semantic gap[18]. Consequently, in the past a lot of research effort has been invested into relevance feedback algorithms. Such algorithms can only be successful if the similarity model of the underlying query engine is flexible enough to represent the user's intention. For example, we use two features f_1 and f_2 and assume a media collection with ten elements. The distribution of the elements in distance space is shown in Figure 1. The query example(s) define(s) the origin. The ground truth of this collection is that 5 elements are similar (shown as *o*) and the 5 other are not (depicted as *x*).

If we use the formula above, we have two parameters that can be manipulated during relevance feedback: the query examples and the weight vector. Anyway, in distance space the result space is always the simplest simplex (a triangle in 2D, a tetrahedron in 3D, etc.), defined by the weights. Thus in the situation above it is impossible to retrieve all similar elements without retrieving the non similar as well. This allows two possible conclusions: (1) the situation above can never occur. Human similarity judgment can never result in such a ground truth with this element distribution or (2) the LWM formula is – as a similarity model – not suitable for VIR. We tend to the second explanation. In the next subsection we will introduce a more flexible model.

3.2 Logical Retrieval

McLuhan writes that images are just illusions while only film (or video) is able to transport visual content appropriately[12]. His statement covers the simple observation that nothing exists without time and that *time means change*. We think that there is deep truth in this statement and derive that similarity measurement based on visual information should not be static but a dynamic process – as it is for human beings.

Logical Retrieval (LR) is based on observation of human behavior. When people are arguing their visual similarity perception they do not do this by making general comments but by pointing out certain aspects and details and stressing their remarkable analogy. These aspects are the features in the querying process. This view of similarity is very old. It

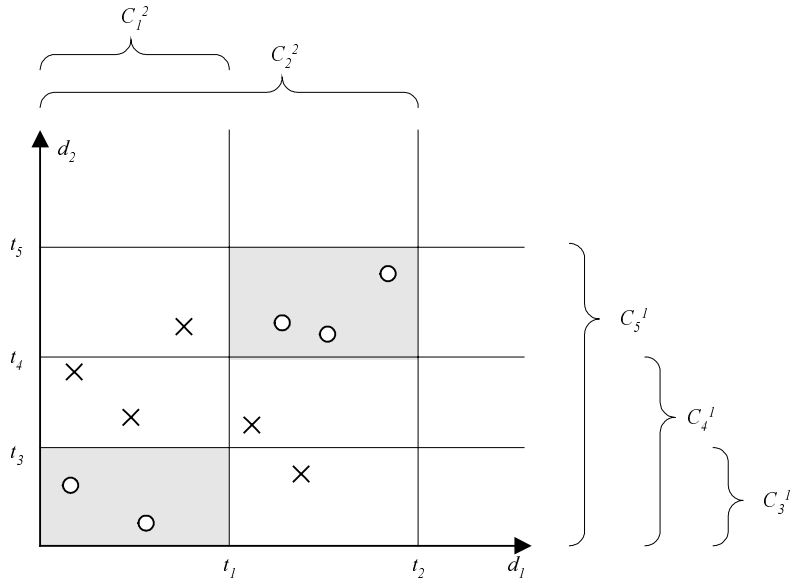

Figure 2: Similarity definition in distance space with logical expressions. The LR approach is a suitable similarity model because it allows every combination of elements. Thus it can represent each possible similarity perception.

was introduced by Aristotle, who saw similarity, when two objects had most or the most important properties, in which they could differ, in common[3]. Derived from this, similarity can be formally defined as the correspondence of objective measurable elements of complex objects or of their physical neighborhood[3]. For example, imagine cartoon figures. The perceived similarity of cartoon figures with real persons comes from extracting and imitating their major features (physique, motion, etc.).

The idea behind LR is simple. It should be a vehicle (model) that allows the selection of every possible combination of elements from a given collection. Thus it does not make any assumption, gives the user full control over the retrieval process and supports every thinkable similarity perception. The standard argument against this technique is: how should the average user be able to handle such a system? To the authors' belief (and experiences) this argument makes little sense at this point, because this is just a user-interface problem, while we are searching for a suitable model for similarity definition. We think that persisting on a very limited model that is easy to handle does not make much sense, if the overall problem of VIR are recall and precision rates of less than 40%.

We define LR similarity measurement as a two-step process[8]. In the first step (micro-level) feature vectors are mapped to points in distance space. Distance space is defined as the vector space that is derived by measuring the distance of media objects to given query examples with distance functions (micro-level similarity measurement). It has one dimension for each unique combination of distance measure and reference stimulus. In the second step (macro-level) the user defines his similarity perception as a logical expression. The logical expression consists of conditions C_i^j of the form given in equation 10. The parameter t_i is a threshold for the maximum distance of a media object for distance space dimension d_j.

$$d_j \leq t_i \tag{10}$$

A media object is added to the result set, if the query expression evaluates to *true* for its distance values. This expression is then refined in an iterative process. We have developed GUI methods where the user need not define the expression directly but implicitly by selecting and moving media objects in a 3D user interface[5]. The set of similar objects in the example above (see Figure 1 in Subsection 3.1) could be described by the following expression (see Figure 2):

$$Query = C_1^2 \wedge C_3^1 \vee \left(C_2^2 \wedge \neg C_1^2\right) \wedge \left(C_5^1 \wedge \neg C_4^1\right) \tag{11}$$

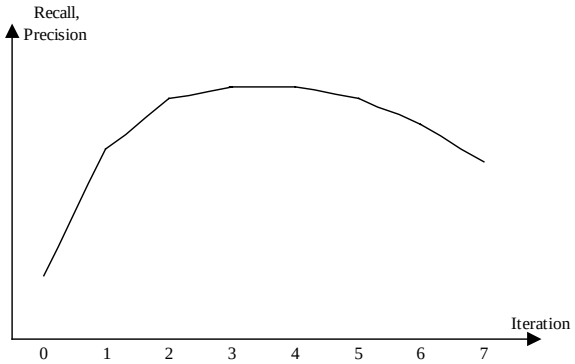

Figure 3: Typical development of recall and precision in iterative refinement by relevance feedback[11]. We propose that – at least partially – this characteristic comes from the user's limited influence on the LWM querying process.

The example makes clear that LR expressions can describe each possible selection in distance space and therefore represent each type of similarity perception. LR is not a new invention. The idea is partially based on logical expressions as in Boolean Retrieval, although it has nothing to do with the Boolean Retrieval method in information retrieval. It is the integration of logical expressions with vector spaces, optimized for visual similarity perception. Our approach to Logical Retrieval is a generalization of our earlier Query Model concept[1]. In addition, we are aware of a second approach that results in a similar concept from a different starting point[13]. Next we will investigate the structure of LR expressions in VIR.

Every query expression can be defined in disjunctive normal form (*OR*-connected terms as in the example above). In this style each *AND*-connected term of an expression describes a n-dimensional cube of elements and should contain at most one condition for each dimension of distance space. Each such n-dimensional cube can be interpreted as a cluster (these clusters correspond to the Query Models we introduced in earlier publications[1]). The Logical Retrieval process is essentially describing clusters of similar elements. According to information retrieval theory, such clusters exist for each collection of elements with reasonable size.

If we interpret LR as disjunctive concatenating of cluster expressions we can do the following simplifications:

1. The *NOT* can be integrated into the cluster terms. Instead of $NOT\, C_i^j$ we can write:

$$NOTC_i^j \equiv \neg\left(d_j \le t_i\right) \equiv d_j > t_i \equiv \overline{C_i^j} \tag{12}$$

2. Two disjunctive conditions on a distance space dimension x in a single cluster term can be integrated into a single expression and written as follows:

$$C_i^x \wedge \overline{C_j^x} = d_x \le t_i \wedge d_x > t_j \equiv t_j < d_x \le t_i \equiv C_{i,j}^x \tag{13}$$

We call $C_{i,j}^{\ x}$ a cluster restriction. Each *AND*-connected expression of cluster restrictions describes an either finite (one cluster restriction for each dimension of distance space) or (ideally) infinite cluster (less cluster restrictions than dimensions).

Next we will point out the major advantages and drawbacks of LR. In opposition to the arguments above, LR has a very positive consequence for user-interface design in VIR. It is intuitive and easy to visualize. Queries are usually represented by selecting example elements. In LR this selection can be made by dragging rectangles around two-dimensional views of examples in feature space or distance space. Views can be created by selecting arbitrary features for the X- and Y-axis. Obviously, these rectangles can be directly transformed into an LR expression. Iterative refinement can be performed in the same way. Abstract annotations like 'very similar' are not needed[5].

By now, it is accepted that iterative refinement based on LWM is limited. It leads to improvements in recall and

precision in the first four to five cycles[11]. Then the quality of the results begins to decrease. Figure 3 describes the typical development of recall and precision over multiple refinement iterations. We think that that the reason for this is the limited influence of the user in a querying process based on LWM. Four to five cycles is exactly the time it takes to adjust the weights for a few features to the optimum values. In LR iterative refinement means finding additional clusters and optimizing their borders. This should at least stretch the typical refinement curve and lead to a higher quality peak in the refinement process.

In addition, LR has a nice side-effect on query execution time. Within each *AND*-connected cluster, the result set of a query is independent from the order of the cluster restrictions. An algorithm that sorts the conditions in a way that those, which sort out most elements and/or use the fastest distance functions, are used first in the querying process, would lead to significant query acceleration. We have presented the design and implementation of such an algorithm[7]. It reduces the average query execution time in our test environment by 66% (in comparison to a QBIC system[9, 10] with the same feature classes and distance functions).

4 INTEGRATION OF THE FEATURE CONTRAST MODEL

In the LR model, we would like to incorporate the ideas of the FCM: asymmetry and non-constant self-similarity. Even though the standard FCM works on binary predicates that are related to the conditions from above (equation 10), we think that that the distinction between symmetric and asymmetric queries belongs to the micro-level and therefore FCM should be incorporated as a (general-purpose) distance measure. To do this, we are not going to interpret numeric feature vector elements fuzzy or probabilistic, because we cannot give good reasons for such an interpretation. Instead, we use the following substitute for continuous data: the similarity function *s()* is defined as in equation 7 and the set operators are replaced by suitable continuous functions. The intersection operator is replaced by one of the two following functions:

$$\text{inter}_c(X,Y) = (a_i) \; where \; a_i = \begin{cases} \dfrac{x_i + y_i}{2} & if \; \max - \dfrac{x_i + y_i}{2} \le \varepsilon_1 \\ 0 & else \end{cases} \tag{14}$$

$$\text{inter}_d(X,Y) = (a_i) \; where \; a_i = \begin{cases} \max - |x_i - y_i| & if \; |x_i - y_i| \le \varepsilon_1 \wedge \max - x_i < \varepsilon_1 \\ 0 & else \end{cases} \tag{15}$$

max is the maximum distance of feature vector elements x_i and y_i and ε_1 approaches *0*. $inter_c$ emphasizes common properties of *X* and *Y* while $inter_d$ emphasizes their differences. In the tests in section 5 we will try to find out which formula performs better for continuous data. For the subtraction operator we use the function from equation 16:

$$\text{sub}(X,Y) = (a_i) \; where \; a_i = \begin{cases} x_i - y_i & if \; \max - (x_i - y_i) \le \varepsilon_2 \\ 0 & else \end{cases} \tag{16}$$

This model should preserve the idea of the FCM. The intersection operator selects properties that are present in both stimuli to a similar extent and the subtraction operator selects properties that are present just in *X*. For *f()* we suggest to use formula 17. The definition of *f()* is not part of Tversky's FCM model. Therefore we do not include it in the continuous model either.

$$f'(X) = \frac{\sum_i a_i}{i} \; where \; a_i = \begin{cases} val(t) & if \; x_i \neq 0 \\ 0 & else \end{cases} \tag{17}$$

t is the determining parameter of *f()*. *val(t)* returns the value of t: the constant's value if t is a constant (e.g. *val(2)=2*) or the variable's value if *t* is a variable (*val(x)=2* if *x=2*). Thus, if *t=1*, *f()* is the cardinality of relevant properties (equivalent to FCM). If $t=x_i$, *f()* measures the mean of the difference of all relevant properties of two stimuli (either both present or only one present).

Two problems are connected to this approach: how to choose ε_1 and ε_2, and how to set the parameters α and β that determine, if the FCM is symmetric or asymmetric. We suggest to base the selection of ε_1 and ε_2 on statistical analysis of the given feature data (with $\varepsilon_i << max$!) and to implement the setting of α and β by a switch in the user interface that allow the specification of symmetric ($\alpha=\beta=0$) and asymmetric queries as well as subtractive ($\alpha>0$ or $\beta>0$) and non-subtractive queries. Below, in Section 5 we will show how this continuous FCM model was implemented in a prototype.

Next we investigate the behaviour of the continuous FCM for binary predicates. Ideally, the continuous FCM should produce the same results for binary predicates as the original FCM. The following tables show all possible relations for two predicate vectors $X=(x_i)$ and $Y=(y_i)$. The intersection (*inter*) should be '1' only if predicate i is present both in X and Y. The subtraction (*sub*) should be '1' if a predicate is present just in X.

x_i	y_i	*inter*	$\max - \frac{x_i + y_i}{2} \le \varepsilon_1$	$inter_c$	$\lvert x_i - y_i \rvert \le \varepsilon_1 \wedge \max - x_i < \varepsilon_1$	$inter_d$
1	1	1	$1-(1+1)/2=0 < \varepsilon_1$ ... *true*	1	$\lvert 1-1 \rvert=0 < \varepsilon_1$ ... *true*, $1-1 < \varepsilon_1$... *true*	1
1	0	0	$1-(1+0)/2=0{,}5 < \varepsilon_1$ ... *false*	0	$\lvert 1-0 \rvert=1 < \varepsilon_1$ ... *false*	0
0	1	0	$1-(0+1)/2=0{,}5 < \varepsilon_1$ ... *false*	0	$\lvert 0-1 \rvert=1 < \varepsilon_1$ ... *false*	0
0	0	0	$1-(0+0)/2=1 < \varepsilon_1$ ... *false*	0	$\lvert 0-0 \rvert=0 < \varepsilon_1$ ... *true*, $1-0=1 < \varepsilon_1$... *false*	0

Table 1. Evaluation of intersection operator for binary predicates.

x_i	y_i	*sub*	$\max - (x_i - y_i) \le \varepsilon_2$	$x_i - y_i$
1	1	0	$1-(1-1)=1 < \varepsilon_2$ ... *false*	0
1	0	1	$1-(1-0)=0 < \varepsilon_2$ ... *true*	1
0	1	0	$1-(0-1)=2 < \varepsilon_2$ ... *false*	0
0	0	0	$1-(0-0)=1 < \varepsilon_2$ ... *false*	0

Table 2. Evaluation of subtraction operator for binary predicates.

For binary predicates $max=1$. If we set $\varepsilon_1<0{,}5$ and $\varepsilon_2<1$ the tables show that all suggested operators perform as desired. That means, if the continuous operators are fed with binary predicates the behaviour of the model is exactly the same as for Tversky's model. This is independent from the selection of $f()$. In the next section we will investigate if the operators are suitable for practical use in VIR systems.

5 TESTS AND RESULTS

Goal of the tests is to measure the performance of the continuous FCM as a *general-purpose* distance measure in comparison to another standard distance measure: the Euclidean distance. The principal superiority of the LR approach over LWM has already been shown in other publications[1, 8, 13]. We have implemented the FCM models from section 4 in a Perl prototype. Perl was chosen because it offers powerful data processing capabilities and allows rapid prototyping. Additionally, by now powerful image analysis libraries exist for Perl.

The selection of the test procedure was problematic. Normally, new VIR methods are tested by selecting a large image library (e.g. the Corel-library), defining a ground truth based on semantic image properties (e.g. images of flowers, images of cars) and evaluating the new method by a reasonably large number of queries with the recall and precision measures[18]. The general problem with this procedure is the following: based on the LR model as a flexible similarity measurement process it is always possible to maximize recall and precision at the same time. Additionally, here we want to measure the performance of FCM as a distance measure on the micro-level. If the performance was weak, the overall system performance in terms of recall and precision could still be good because of LR's flexibility. Especially, the characteristic advantages of FCM cannot be measured with such a procedure.

Because of these considerations we gave up the idea of an evaluation based on recall and precision and developed the following test pattern. We compare the cluster structure of the distance spaces created on the micro-level by the used distance measures. A cluster is defined as a group of objects that belong to the same semantic group as the query example (defined by the ground truth). In detail we are doing the following (in a reasonable number of repetitions).

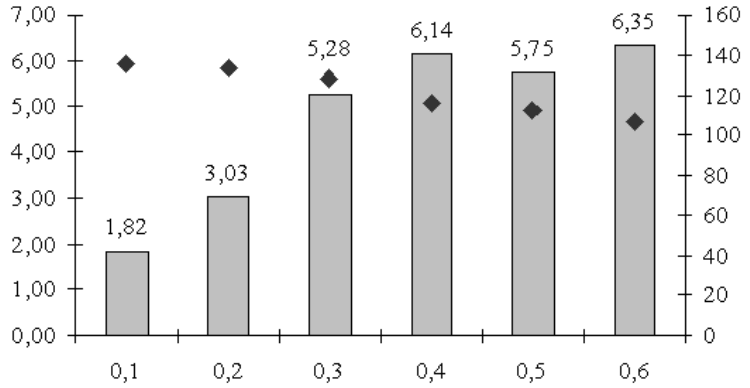

Figure 4: Average cluster size (bars, left Y-axis) and cluster pollution (rhombs, right Y-axis, in percent of correct cluster members) for the symmetric non-subtractive FCM ($\alpha=\beta=0$) depending on ε_1 (X-axis).

Based on a given set of feature vectors and a randomly selected query example we generate two distance spaces: one with FCM and the other with Euclidean distance. In these spaces we identify all clusters of objects that belong to the group of the query example and calculate the average group size and the group size variance. Average and variance of the group size are meaningful measures for the quality of a distance function in LR, because a good distance measure should generate as large as possible (and therefore as less as possible) clusters that can then be easily tracked on the macro-level. The major problem of this approach is the identification of clusters in an n-dimensional distance space (derived from n features). This is non-trivial but can be avoided by taking all features together and measuring the distance as a *whole*. The generated distance space is then one-dimensional and the test for the general-purpose distance measures is even harder (especially for the FCM) because they have to integrate arbitrarily related features.

For the tests we use a collection of 444 images of coats of arms. The images are synthetic (painted, not scanned or photographed) and have been described in earlier publications[1, 2]. Like Santini and Jain[15] we think that computer-based similarity assessment should be pre-attentive and therefore VIR benchmarks should be based on pre-attentive similarity judgement as well. This can be achieved by using media collections with *abstract* content for evaluation. Unfortunately, we are not aware of any visual media collection with really abstract content. Therefore we think that using the coats of arms library instead is a good compromise, because coats of arms carry no inherent visual meaning. Even though the elements of arms have precisely defined semantics the visual image itself has no meaning at all (except some ordinaries like horses, crowns, etc.). To select query examples and identify clusters we need a ground truth. Based on the visual impression we built a pre-attentive ground truth of four groups of images with similar colors, layouts and textures. The group size varies from 18 to 24 images. Finally, for the distance calculation we need feature vectors. We use the features from our coats of arms CBIR system[1, 2]. These include color histograms (global and localized) and other color features, object features (contours, etc.), image symmetry features and application-specific features (coats of arms segmentation, etc.). Each of the 444 images is represented by a feature vector with 58 elements.

Testing FCM as a general-purpose distance measure we want to clarify the following question: are the characteristics of the FCM as a tailor-made similarity measure still relevant in the LR model? Our hypothesis is: yes. We try to answer this question with three tests: (1) Performance comparison of FCM with the $inter_c$ intersection operator to FCM with the $inter_d$ operator, (2) comparison of symmetric FCM without consideration of features that are only present in one stimulus ($\alpha=\beta=0$, non-subtractive) to asymmetric and/or subtractive FCM, and (3) comparison of the best FCM model to the Euclidean distance. To optimize FCM the optimal values for the parameters t, α, β, ε_1 and ε_2 have to be found. In summary we run 217000 queries: 1000 on Euclidean distance (no parameters), 72000 on FCM with $inter_d$ (6 parameters) and 144000 on FCM with $inter_c$ (4 parameters, $f()$ with $t=1$ was not evaluated, see below).

The comparison of $inter_c$ and $inter_d$ lead to very clear results. FCM with $inter_c$ (emphasizes common features) was in every single test better than FCM with $inter_d$ (emphasizes differences) with equal parameters. While the average number of clusters for FCM with $inter_c$ is most times less than 9 elements it is nearly always higher than 9 for FCM with $inter_d$ intersection operator. That means, in average the objects of the query examples group fall in more than 9 clusters in distance space. Consequently, $inter_d$ was not considered in the rest of the evaluation. Next we tried to

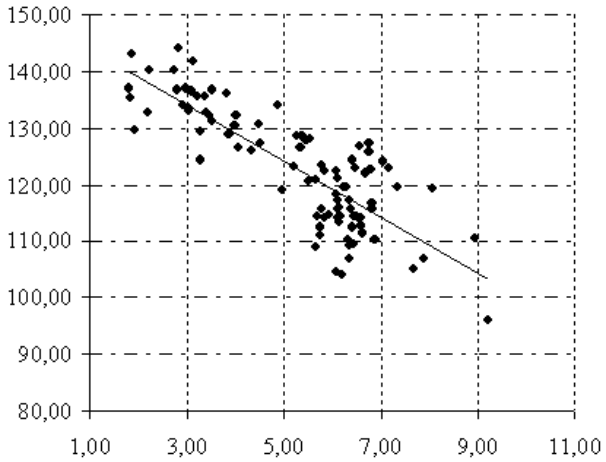

Figure 5: Correlation of average cluster size (X-axis) and average cluster pollution (Y-axis) for symmetric non-subtractive FCM. The correlation is not significant (81,9%).

optimize the parameter t of FCM and found out the following: $t=x_i$ clearly outperforms $t=1$. That means, function $f()$ with continuous predicate interpretation is always better than $f()$ with binary interpretation (and FCM with equal parameters). Thus $t=1$ was not considered in the further tests as well.

To find out whether symmetric non-subtractive FCM is better than asymmetric and/or subtractive FCM we optimized the parameters for each of the four possible combinations. This revealed that in average symmetric non-subtractive FCM performs best. Figure 4 shows the results for $\alpha=\beta=0$. The bars show the average number of clusters in distance space depending on ε_1 while the rhombs show a new phenomenon that we call cluster pollution. First, we concentrate on the average number of clusters. We can see that it decreases with decreasing ε_1 from 6 (about constant for $\varepsilon_1 > 0,3$) to about 2 ($\varepsilon_1 = 0,1$). This is because if ε_1 is set smaller, less predicates are used to judge the similarity of objects. From the small number of clusters we can conclude that FCM has an inherent 'intelligence' to select the *right* properties and using lower epsilons results in a better cluster structure.

Cluster pollution means that in a cluster of adjacent objects from the queried group, false objects exist that have *exactly* the same distance value as one of the cluster members but do not belong to the clustered group (according to the ground truth). Such false objects cannot be identified with LR expressions and therefore have to be treated as cluster members. Generally, it should be very unlikely that two objects come out at exactly the same point in distance space but because of the nature of FCM (only some predicates are used, controlled by ε_1) this can happen. In Figure 4 we see that cluster pollution decreases with increasing ε_1 from 140% to 100%. That means for $\varepsilon_1 = 0,6$ each cluster contains about the same number of correct and false members. Of course, to a large degree this can be explained by the one-dimensional distance space. If distance space had more dimensions the clusters would be less polluted. Still, cluster pollution is a consequence of using FCM. Clusters in an Euclidean distance space are not polluted (see below). The results in Figure 4 may suggest that a lower number of clusters (gained by lowering ε_1) corresponds with higher cluster pollution. For clarification we calculated the correlation of average cluster size and cluster pollution. Figure 5 shows the results. There is no significant correlation between the cluster size and cluster pollution. The correlation coefficient is lower than 82%.

The results for asymmetric and/or subtractive FCM can be seen in Figure 6. In this case, either α, β or both are greater 0 and therefore the results depend on ε_1 and ε_2. The left diagrams show the average number of clusters. Black areas (combinations of ε_1 and ε_2) mark results of average 2 clusters (1,5-2,5 clusters of correct objects in distance space), dark grey results of average 3 clusters (2,5-3,5), and so on. The right diagrams show the average cluster pollution. Black areas have a cluster pollution of average 135%, dark grey of 130%, et cetera. The first row of diagrams shows the results for FCM with $\alpha=1$ and $\beta=0$. In this case all features are taken into account that exist in both objects or only in the query example. The second row of diagrams shows the results for FCM with $\alpha=0$ and $\beta=1$. These two FCM configurations are asymmetric and subtractive. The third row of diagrams shows the results for FCM with $\alpha=1$ and $\beta=1$. This configuration is symmetric and subtractive. We have only investigated these cases but not linear combinations between them, because these configurations are extreme cases and the results of intrapolated

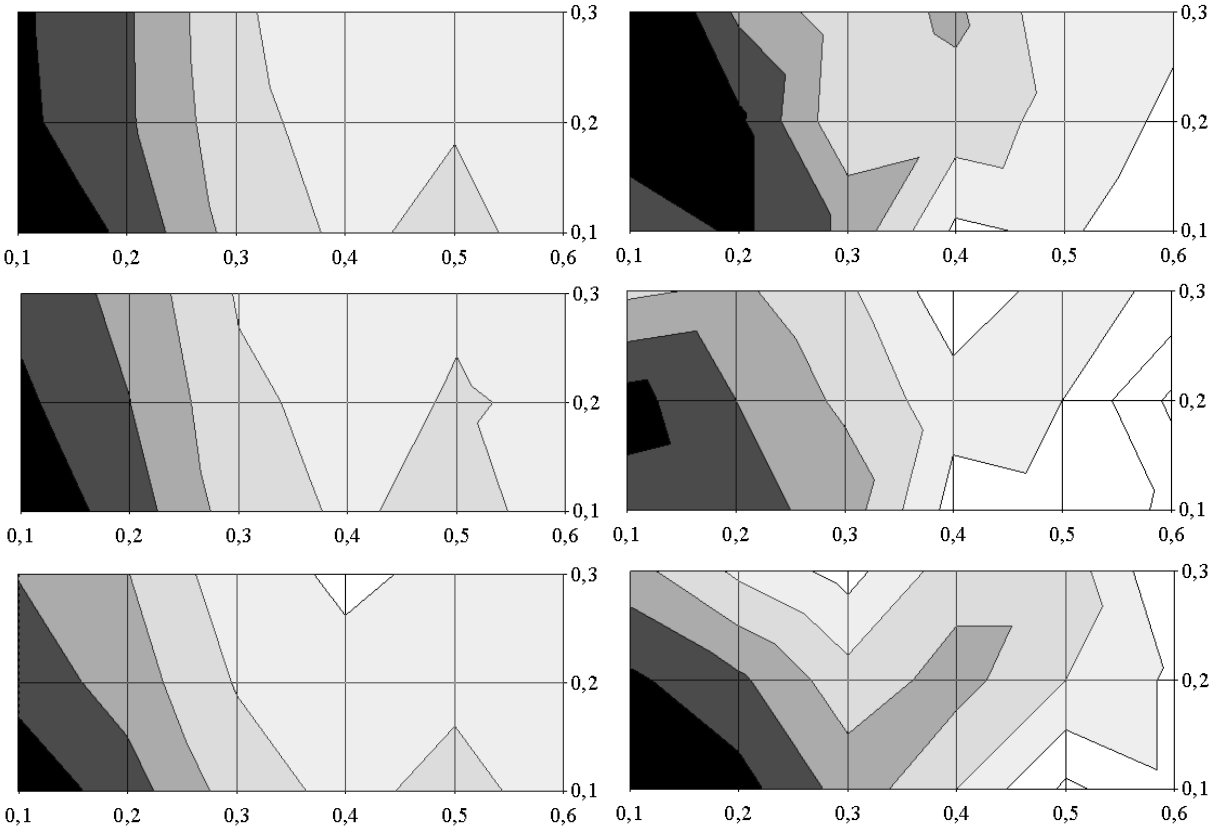

Figure 6: Cluster size (left column) and cluster pollution (right column) depending on ε_1 (X-axis) and ε_2 (Y-axis).
First row: $\alpha=1$, $\beta=0$, second row: $\alpha=0$, $\beta=1$, third row: $\alpha=1$, $\beta=1$.

configurations would be just intrapolations of the extreme results.

From the diagrams we can induce the following observations: the FCM with $\beta=0$ is in the average better than the FCM with $\beta=1$. That means, using the information of features that are only present in the compared object Y does not improve the results. For visual information this is intuitive: non-similar objects can have a vast number of different features. Using them in a query is misleading and can hardly be supported by arguments. Additionally, we can see from the first row of images that for very low epsilons there is an area with a lower cluster pollution. This supports our argumentation that bigger clusters are not just a trade-off for higher cluster pollution.

In comparison to Euclidean distance (applied to the same test data) we see that FCM performs much better. Euclidean distance generates clusters of about two members. That means in average the queried image collection in distance space falls in 10 clusters. This is much worse than for FCM (2-8 clusters). On the other hand, Euclidean distance has two advantages: the resulting clusters have no pollution (because all features of an object are taken into account for distance measurement) and Euclidean distance is faster than FCM. The relationship in query execution time for the same test data on the same system is about 3:1. That means an FCM query takes about 3 times as long as an Euclidean query.

6 CONCLUSION

This paper reviewed similarity models for Visual Information Retrieval (VIR) and introduced the Feature Contrast Model (FCM) for VIR distance measurement. In the first part underlying concepts of similarity modeling were revisited and the standard VIR model was sketched. Motivated by the drawbacks of this model the more flexible Logical Retrieval model (LR) was introduced. According to this model, distance measurement is reduced from the central element of similarity measurement to a less important role. It should help to organize similar objects in distance space in a way that they are easy to find in an iterative querying process. In the second part the FCM, developed by

psychologists to explain human peculiarities in similarity perception, was integrated in LR as a general-purpose distance measure. To do that a continuous model of FCM was developed. This model was tested on an abstract image library with a pre-attentive ground truth to judge its performance and find out the optimal parameterization.

The results show that FCM performs (in the LR context) better than Euclidean distance. Euclidean distance was used for comparison because it is used in many VIR systems and is based on the (questionable) metric axioms. FCM minimizes the number of clusters in distance space. Therefore it is the ideal distance measure for LR. FCM allows a number of different parameterizations. The tests revealed that in the average a symmetric, non-subtractive configuration that emphasizes common properties of visual objects performs best. Its major drawback in comparison to Euclidean distance is its worse performance (in terms of query execution time).

In future work we will try to improve the performance of FCM. Additionally, we will develop heuristics for FCM configuration for various kinds of feature data (setting ε_1, ε_2 and α). To do this, we will integrate FCM in the VIR project VizIR. VizIR aims at developing an open framework for VIR[6]. Interested researchers are invited to contact the authors for more information.

7 REFERENCES

1. C. Breiteneder, H. Eidenberger, "A Retrieval System for Coats of Arms", *Proceedings International Symposium on Multimedia Application and Distance Education*, Baden-Baden, 1999 (available from http://www.ims.tuwien.ac.at/~hme/papers/isimade1999.pdf).
2. C. Breiteneder, H. Eidenberger, "Content-based Image Retrieval of Coats of Arms", *Proceedings IEEE International Workshop on Multimedia Signal Processing*, 91-96, IEEE, Helsingör, 1999.
3. W. Butollo, *Subjective and Objective Similarity in Verbal Learning*, Notring Verlag, Vienna, 1968 (in German).
4. A. Del Bimbo, *Visual Information Retrieval*, Morgan Kaufmann Publishers, San Francisco , 1999.
5. H. Eidenberger, C. Breiteneder, "A Framework for User Interface Design in Visual Information Retrieval", *Proceedings IEEE International Symposium on Multimedia Software Engineering*, IEEE, Newport Beach, 2002.
6. H. Eidenberger, C. Breiteneder, "A Framework for Visual Information Retrieval", *Proceedings Visual Information Systems Conference*, 105-116, Springer Verlag, HSinChu 2002.
7. H. Eidenberger, C. Breiteneder, "Performance-optimized feature ordering for Content-based Image Retrieval", *Proceedings European Signal Processing Conference*, EUSIPCO, Tampere, 2000 (available from http://www.ims.tuwien.ac.at/~hme/papers/eusipco2000.pdf).
8. H. Eidenberger, C. Breiteneder, "Visual Similarity Measurement in VizIR", *Proceedings IEEE Multimedia Conference*, IEEE, Lausanne, 2002.
9. M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, P. Yanker, "Query by Image and Video Content: The QBIC System", *IEEE Computer*, **28/9**, 23-32, 1995.
10. IBM QBIC Website, http://wwwqbic.almaden.ibm.com/, last visited: 18th October 2002
11. C. Leung, "Visual Information Search and Benchmarking", *Visual Information Systems Conference*, Springer Verlag, HSinChu, 2002 (Plenary Talk).
12. M. McLuhan, *Understanding Media*, McGraw-Hill Publishers, New York, 1964.
13. M. Ortega, R. Yong, K. Chakrabarti, K. Porkaew, S. Mehrotra, T.S. Huang, "Supporting Ranked Boolean Similarity Queries in MARS", *IEEE Transactions on Knowledge and Data Engineering*, **10/6**, 905-925, 1998.
14. Y. Rui, T.S. Huang, M. Ortega, S. Mehrotra, "Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval", *IEEE Transactions on Circuits and Systems for Video Technology*, **8/5**, 644-655, 1998.
15. S. Santini, R. Jain, "Similarity is a Geometer", *Multimedia Tools and Applications*, **5/3**, 277-306, 1997.
16. S. Santini, R. Jain, "Similarity Matching", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **21/9**, 871-883, 1999.
17. P.P. Sint, *Similarity Structures and Similarity Measures*, Austrian Academy of Sciences Press, Vienna, 1975 (in German).
18. A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain, "Content-based image retrieval at the end of the early years", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22/12**, 1349-1380, 2000.
19. J.R. Smith, S.F. Chang, "VisualSEEk: a fully automated content-based image query system", *Proceedings ACM Multimedia*, 87-98, ACM Press, Boston, 1996.
20. A. Tversky, "Features of Similarity", *Psychological Review*, 84/4, 327-352, 1977.