

Toward Complete Performance Characterization in Content-based Retrieval

Nicu Sebe^a, Nies Huijsmans^b, Qi Tian^c, and Theo Gevers^a

^a Faculty of Science, University of Amsterdam, The Netherlands;

^bLeiden Institute of advanced Computer Science, Leiden University, The Netherlands;

^cUniversity of Texas at San Antonio, USA

ABSTRACT

In Content-based Image Retrieval the comparison of a query image and each of the database images is defined by a similarity distance obtained from the two feature vectors involved. These feature vectors can be seen as sets of noisy indexes. Unlike text matching (that is exact) image matching is only approximate, leading to ranking methods. Only images at the top ranks (within the scope) are returned as retrieval results. Image retrieval performance characterization has mainly been based on measures available from probabilistic text retrieval in the form of Precision-Recall or Precision-Scope graphs. However, these graphs offer an incomplete overview of the image retrieval system under study. Essential information about how the success of the query is influenced by the size and type of irrelevant images is missing. Due to the inexactness of the visual matching process, the effect of the irrelevant embedding, represented in the additional performance measure generality, plays an important role. In general, a performance graph will be three-dimensional, a Generality-Recall-Precision Graph. By choosing appropriate scope values a new two-dimensional performance graph, the Generality-Recall=Precision Graph, is proposed to replace the commonly used Precision-Recall Graph, as the better choice for total recall studies.

Keywords: performance characterization, content-based retrieval, generality, precision-recall graphs

1. IMAGE SIMILARITY AND RANKING METHODS

Content-based Image Retrieval differs from text retrieval in that exact matches are not the main aim; one would rather like to retrieve images that are sufficiently similar in their visual characteristics or image derived features. This is reflected in the feature vectors used for image comparison that are to be considered rather noisy indexes to the underlying images they represent. During image queries, the feature vector of a search image is compared to each feature vector of the database images and the difference between the feature vector elements is condensed into a similarity distance. Given these similarity values, all the images in the database can be ranked according to the query image, and the top of this ranking list (all images up to a certain scope) are returned by the system as retrieval results. How to select an appropriate scope value is a yet unsolved question; usually a standard scope is used. Another circumstance that differs between text matching and image matching is the role of the embedding irrelevant images in the database. Whereas in exact sub-string matching the number of irrelevant text documents does not matter, in approximate image feature matching it does matter how the relevant items of the query database are embedded into what sort and what number of irrelevant items. In this paper, we will indicate how the influence of this irrelevant embedding can be made explicit by using an additional measure, called generality, that represents the fraction of relevant material in the database.

The motivation for this work arose from the experiments carried out on our Leiden 19th-Century Portrait Database (LCPD). This database at present contains 21,094 photographic portraits with as many studio logos (on the separately scanned backsides); almost 15,000 of these logos have at least one companion in the database. These logos are manually grouped into 1,856 non-overlapping logo classes with an average size of almost 8 relevant unordered items; relevant class size is in the range [2, 308]. A more extensive description of this ground-truth test set can be found in.¹ Each relevant logo image in turn was used to perform a content-based query by example search. To characterize the outcomes of these similar logo queries, we condensed all the system parameters that precede ranking into Ranking Method. To determine the effectiveness of the ranking methods one needs to select threshold values for the number of top ranked images that will be returned as retrieval result; this threshold or scope is usually just a convenient number of top ranked images, but it always has an

associated (dis)similarity threshold distance in feature space. The success-rate of this scope based similarity query engine can be characterized by recall, being the fraction of all relevant images returned, and/or precision, being the fraction of relevant images within the scope. These measures are dependent as we will show later in this paper, and are both closely linked to the local density values of relevant and irrelevant feature vectors in the corresponding feature space used to index the images.

In Figure 1 we present the performance of a specific ranking method (gray-level images characterized by their gray-level histogram as feature vector and using histogram intersection to obtain a similarity measure for ranking) at a range of scopes and within a growing number of irrelevant logos. The effect of changing the scope is shown by the well-known Precision-Recall curve (PR-curve); the effect of changing the embedding is indicated by the corresponding generality value g of each PR-curve. Without an embedding the retrieval system is as effective as an ideal retrieval system as can be seen from the PR-curve at $g = 1.0$; precision remains 1.0 until all items are retrieved (at recall=1.0). At the second highest curve with $g = 0.2$, meaning that there are 50 times more irrelevant items than relevant ones, the PR-curve already starts to deteriorate; already some irrelevant images happen to have feature vectors that have a higher similarity value when compared to the search image than some of the relevant ones. Finally in the lower left PR-curve with $g = 0.00025$, when there are 4,000 times more irrelevant items than relevant ones, the performance of this gray-level histogram ranking method has dropped to unacceptable low levels.

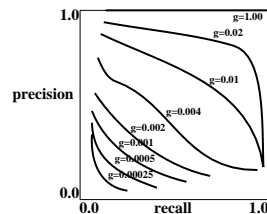


Figure 1. Typical Precision-Recall curves for retrieval of a constant size relevant class of 8 items embedded in a growing number of irrelevant items (max 32,000) using a ranking method based on gray-level histogram intersection: the relevant fraction or generality for each curve is given by its g value.

This effect, of a growing irrelevant embedding on the performance of approximate similarity matching, is due to the inexactness of matching visual indexes obtained from scanned digital images. We therefore see a role for *generality* in characterizing the effect that a vastly growing embedding might have on a final scaled up version of the prototype content-based image search system.

In general, the size of a dynamically growing database might result in a relative growth that is equal for both relevant items and irrelevant embedding items. In this case, the PR graph would remain at the same generality level. Moreover, a constant retrieval recall rate would mean that the scope, used to retrieve these relevant items, would have to increase with the same percentage as well. Hence, it would be advantageous to couple the scope to the size of the relevant class.

2. SHORTCOMINGS OF PERFORMANCE MEASURES

Performance characterization of content-based retrieval often borrows from performance figures developed over the past 30 years for probabilistic text retrieval. Landmarks in the text retrieval field are the books by Salton² and van Rijsbergen³ as well as the proceedings of the annual ACM SIGIR and NIST TREC conferences.

In probabilistic text retrieval like in van Rijsbergen,³ the NIST TREC,⁴ and MPEG-7 descriptor performance evaluation,⁵ authors often go for single measure performance characterizations. These single measures are based on ranking results within a limited scope without taking into account both the size of the relevant class and the effect of changing either the size or the nature of the embedding irrelevant items. By their nature, these single measures have limited use, because their value will only have a meaning for standardized comparisons, where most of the retrieval parameters, such as the embedding, relevant class size, and scope are kept constant.

In the area of probabilistic retrieval, the results of performance measurements are often presented in the form of Precision-Recall (or Recall-Precision) graphs and Precision-Scope graphs. Each of these standard performance

graphs provides the user with incomplete information about how the Information Retrieval System will perform for various relevant class sizes and various embedding sizes. Generality (influence of the relevant fraction) as a system parameter hardly seems to play a role in performance analysis.⁶⁻⁸ Although generality may be left out as a performance indicator when competing methods are tested under constant generality conditions, it appears to be neglected even in cases where generality is widely varying (a wide range of relevant class sizes in one specific database is the most frequently encountered example). A continually growing embedding, around a class of relevant items, will normally lower the overall *precision, recall* curve to unacceptable low levels as it is evident from average query results obtained at a range of specific *generality* conditions (see Figure 1).

The lack of generality information, in Precision-Recall and Precision-Scope graphs, makes it difficult to compare different sized IR Systems and to find out how the performance will degrade, when the irrelevant embedding is largely increased. The recent overview by Müller et al.⁹ does not even mention generality as one of the required parameters for performance evaluation.

These considerations led us to re-evaluate the building blocks of information retrieval performance measurement for the area of Content-based Image Retrieval (CBIR) and the way these performance measures are visualized in graphs.¹⁰ How can we make the performance measures for image queries on test databases complete, so that results of specific studies can not only be used to select the better method, but can also be used to make comparisons between different system sizes and different domains? In the next section, we argue that the present single measures, and in particular the Precision-Recall Graph, are not only unsuited for comparing different systems but are often also flawed in their use of averaging over various relevant class sizes and embedding ratios.

3. PERFORMANCE EVALUATION AS A DECISION SUPPORT ANALYSIS

In Information Retrieval (IR) the user having specified a query would like the system to return some or all of the items (either documents, images, or sounds) that are in some sense part of the same semantic relevant class i.e., the relevant fraction of the database with respect to this query for this user at this time.

In a testing environment, the performance of the Retrieval System, in its selection of database items that are retrieved, should be compared to the equivalent situation where ground-truth has been constructed. An Ideal Information Retrieval System would mimic this ground-truth. Such an Ideal IR System would quickly present the user some or all of the relevant material and nothing more. The user would value this Ideal System as being either 100% effective or being without (0%) error. In this paper, we will refer to this Ideal System as the Total Recall Ideal System (TRIS). In practice, however, IR Systems are often far from ideal: generally the query results shown to the user (a finite list of retrieved elements) are incomplete (containing only some retrieved relevant class items) and polluted (with retrieved but irrelevant items).

We characterize a CBIR system using the following set of parameters:

$$\text{number of relevant items for a particular query} = \text{relevant class size} = c \quad (1)$$

$$\text{number of irrelevant items for a particular query} = \text{embedding size} = e \quad (2)$$

$$\text{ranking method} = m \quad (3)$$

$$\text{number of retrieved items from the top of the ranking list} = \text{scope} = s \quad (4)$$

$$\text{number of visible relevant items within scope} = v \quad (5)$$

$$\text{total number of items in the ranked database} = \text{database size} = (c + e) = d \quad (6)$$

In this set-up the class of relevant items is considered unordered and everything that precedes a particular ranking (like user feedback) is condensed into the *ranking method*. Performance is determined by the particular combination of the 4 free parameters, since the relevant outcome of a particular query, v , is a function of class size c , embedding size e , ranking method m , and scope s . In general, however, the average performance will be graphed for a number of ranking methods, to completely specify the retrieval system performance for a ground checked set of queries:

$$v = v_m = f(c, d, s). \quad (7)$$

We will concentrate on retrieval settings where the embedding items vastly outnumber the relevant class items, $e \gg c$ and hence $d \approx e$.

In our opinion, a characterization of the Retrieval System performance should be based on the well-established decision support theory similar to the way decision tables or contingency tables are analyzed by Gokhale and Kullback.¹¹ From a quantitative decision-support methodology our Query By Example (QBE) situation can be characterized for each ranking method by a series of decision tables or, as they are also called, contingency tables.¹¹ A decision table for a ranking method represents a 2×2 matrix of (relevant, irrelevant) versus (retrieved, not retrieved) number of items for different choices of scope s , relevant class size c , and embedding e . It can also be seen as the database division according to the ground-truth versus its division according to Content-Based Information Retrieval (CBIR) at specific scope. The CBIR preferred choice of contingency table descriptors is given next to the Decision Support naming scheme in Table 1.

v	$(c - v)$	c	TP	FN	P
$(s - v)$	$(d + v) - (c + s)$	e	FP	TN	N
s	$(d - s)$	d	R	NR	DB

Table 1. Categories and marginals for the contingency tables where P = Positive, N = Negative, FP = False Positive, FN = False Negative, TP = True Positive, TN = True Negative, R = Retrieved, NR = Not Retrieved, DB = Database size. In TRIS $v = s = c$ and $TP = P = R$.

In general the aim is to minimize a weighted combination of False Positives and False Negatives:

$$\min(\alpha FP + (1 - \alpha)FN) \text{ with } \alpha \in [0.0, 1.0] \quad (8)$$

3.1. Normalized Performance Measures

The performance or relevant outcome of the query, v from Eq. (7), can be normalized by division through either c , s , or d :

$$v/c = \text{recall} = r = f(1, d/c, s/c) = f(d/c, s/c) \quad (9)$$

$$v/s = \text{precision} = p = f(c/s, d/s, 1) = f(c/s, d/s) \quad (10)$$

$$v/d = f(c/d, 1, s/d) = f(c/d, s/d) \quad (11)$$

$$\text{with } c/d = \text{generality} = g = \text{expected random retrieval rate} \quad (12)$$

Eq. (11) is not very useful in this form since both a low v and a high d will result into low performance figures (especially in our case of $d \approx e \gg c$).

Recall and *precision* are widely used in combination (Precision-Recall graph) to characterize retrieval performance usually giving rise to the well-known hyperbolic graphs from high *precision*, low *recall* towards low *precision*, high *recall* values. *Precision* and *recall* values are usually averaged over precision or recall bins without regard to class size, *scope*, or embedding conditions. That these are severe shortcomings can be seen from Eqs. (9) and (10) where *recall* and *precision* outcomes are mutually dependent and may vary according to the embedding situation. How the dependency of *precision* and *recall* restricts the resulting outcomes is described in Section 3.3; how it affects the way p, r value pairs should be averaged is described in Section 3.4. In the next section, we will further normalize the performance description resulting in measures that are all normalized with respect to the relevant class size c and retain information about the effect of a vastly growing embedding e .

3.2. Additional Normalization of Scope

Remembering TRIS, the ideal total recall system introduced before, and observing the ratios in Eqs. (9) and (10), we propose to further normalize performance figures by restricting scopes to values that have a constant ratio with respect to the class sizes involved:

$$s_r = \text{relevant scope} = \frac{\text{scope}}{\text{relevant class size}} = \frac{s}{c} = \frac{r}{p} = a = \text{constant} \quad (13)$$

With this relevant scope restriction, Eqs. (9) and (10) become:

$$r = f(1, d/c, ac/c) = f(1, d/c, a) = f(d/c) \quad (14)$$

$$p = r/a = f(c/ac, d/ac, 1) = f(1/a, d/ac, 1) = f(d/c). \quad (15)$$

This additional normalization of *scope* with respect to class size c means that the degrees of freedom for performance measures are further lowered from 2 to 1; only *recall* or *precision* values have to be graphed versus an embedding measure. Our preferred choice for the constant a in Eq. (13) is to set $a = 1$ (this corresponds to setting $\alpha = 0.5$ in Eq. (8)). With this setting one actually normalizes the whole Table 1 (now with $s = c$) by c , thus restricting ones view to what happens along the diagonal of the Precision-Recall Graph where $p = r$. This view coincides to a comparison of the retrieval system under study with TRIS (see Section 3.3).

The only remaining dependency in this set-up (apart from the method employed) is on d/c . In Eq. (12) its inverse was defined as *generality* or the expected success-rate of a random retrieval method. Although generality g is a normalized measure, we will not graph it as such, because this would completely obscure the performance behavior for our case of interest, a range of $e \approx d \gg c$. Instead we propose to graph $p = r/a$ versus $-\log_2(g)$ or $\log_2(d/c)$ to make the generality axis unbounded by giving equal space to each successive doubling of the embedding with respect to the relevant class size.

3.3. Scope Graphs Contained in P-R Graphs: Normalized Scope

Information about the effect of changing the *scope* on the measured *precision* and *recall* values can be made visible in the Precision-Recall graph by taking into account that possible *precision*, *recall* outcomes are restricted to lay on a line in the PR-graph going through the origin. This is due to the fact that the definitions of *precision* and *recall* have the same numerator and are therefore not independent. The dependent pair of p, r values, and its relation to scope, becomes even more pronounced when scope is normalized with respect to the number of relevant items as defined by Eqn 13. We called this measure, *relevant scope*, and present p, r values accompanied by their relevant scope line (radiating from the origin). So for each scope $s = a \cdot c$ with a an arbitrary positive number, $s_r = a$ and the p, r values are restricted to the line $p = r/a$. Therefore, the scope information can be displayed as a radiating set of lines from the origin of the PR graph. In Figure 2(a), a number of constant scope lines for retrieval of a relevant class of four additional relevant class members is shown.

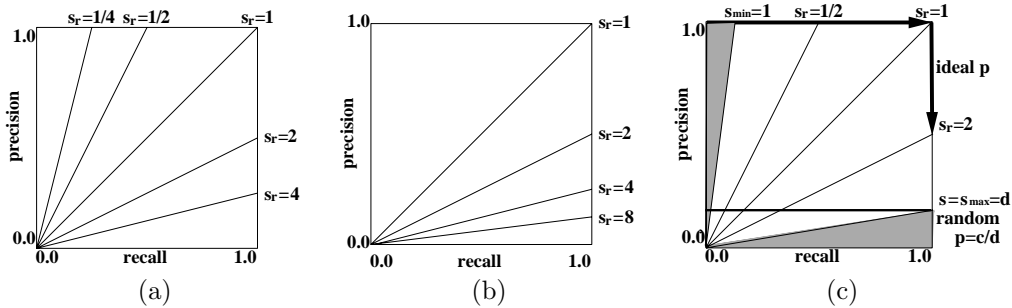


Figure 2. (a) Lines along which p, r values are located at relevant class size=4 for several scopes; (b) Lines along which p, r values for retrieved relevant class size=1 (relevant class of 2, 1 used for query, max 1 for retrieval) are located; (c) p, r values for ideal retrieval are $1, r$ for $r < 1$; for scope size $>$ relevant class size p drops slowly towards random level c/d .

With these relevant scope lines drawn in the Precision-Recall graph one understands much better what the p, r values mean. In the ideal case (see Figure 2(c)), *precision* p will run along $p = 1.0$ for *recall* $r \in [0.0, 1.0]$ and reach $p, r = 1.0, 1.0$ (the TRIS point) when *scope* equals relevant class size ($s = c$); for scopes greater than relevant class size, *precision* will slowly drop from $p = 1.0$ along $r = 1.0$ until the random level $p = c/d$ at $s = d$ is reached. Also depending on relevant class size the region to the left of $p = r/c$ cannot be reached (solving the difficulty in PR-graphs for selecting a precision value for *recall* = 0.0) as well as the region below $p = dr/c$. This means that for the smallest relevant class of 2 members, where one of the relevant class members is used to locate its single partner, the complete upper-left half of the PR graph is out of reach (see Figure 2(b)).

Because the diagonal $s = c$ line presents the hardest case for a retrieval system (last chance of precision being max 1.0 and first chance of recall being max 1.0), and is the only line that covers all relevant class sizes (see Figure 2(b)), the best total recall system performance presentation would be the $p = r$ plane in the three-dimensional GR_eP Graph (Generality-Precision-Recall Graph).

In conclusion, when ground-truth is available and therefore relevant class sizes are known, there is no need for Precision- or Recall-Scope graphs. The diagonal of the PR graph has a special meaning; it is along the diagonal that *precision, recall* values will lay when scopes are equal to the size of the relevant class. One could see the PR graph as a combination of Precision-Scope and Recall-Scope graphs fanning out from its origin. Thus, one can easily interpret PR-graphs, by taking notice of the fact that a line through a *precision, recall* value and the origin is directly related to the scope. The PR-Graph is a sort of polar coordinate way of looking at the Precision-Scope or Recall-Scope Graphs, the angle of the $0, 0 - p, r$ line indicating the relevant scope.

3.4. On Average Precision, Recall Values

For system performance one normally averages the discrete sets of precision and recall values from single queries by averaging *precision, recall* values to obtain the well-known, monotonically decreasing, average PR curves, without paying attention to the generality or scope values associated with those measurements. To compensate for the effect different generality values have on the outcome of the averaging procedures, different ways of averaging are applied, like the micro- and macro-averaging used in text-retrieval.¹² Because *generality* values vary with queries, one can thus obtain averages that equally weigh documents or queries. In the critical review, Raghavan et al.¹³ state, with respect to averaging precision and recall values within the same database, that precision values should be averaged by using constant scope or cut-off values, rather than using constant recall values.

The fact stressed in Section 3.3, that p, r results have associated generality and relevant scope values, also has implications for the way average PR curves should be made up. Instead of averaging precision values within recall or scope bins, one should average precision values along constant scope lines and one should only average these p, r values that share a common generality value. When averaging for query results, obtained from a constant size test database, the restriction to averaging over outcomes of queries with constant relevant class sizes (constant generality value), will automatically result in identical micro- and macro-averages. The view expressed by Raghavan et al.¹³ should therefore even be refined. Their recipe, of averaging measured *precision, recall, scope* values over constant scope values only, should further be refined to our recipe of averaging *precision, recall, relevant scope, generality* values over constant *relevant scope, generality* values only.

An example of the way we obtain an average p, r curve out of 3 individual queries with different relevant scope and a constant generality value in the same embedding is given in Figure 3 (a).

3.5. A Unified View on Total Recall Performance with Generality and Relevant Scope

Because we understand the aversion of switching from two-dimensional to three-dimensional performance graphs, research was done to select attractive two-dimensional cutting planes able to represent system performance for specific needs. In the light of our user's Total Recall Ideal System, introduced earlier, one can highlight the system performance by restricting to the diagonal plane in Generality-Recall-Precision space that contains the *precision, recall* values where relevant scopes are 1.

The two-dimensional Generality-Recall=Precision Graph, showing $recall = precision$ values as a function of *generality* (on a logarithmic scale), will be called the GR_iP Graph for short (see Figure 3 (b) and (c)). For Total Recall studies, one could present several GR_iP related graphs for planes in the GR_eP Graph, where $recall = n \cdot precision$, corresponding to the situation where the scope for retrieval is a multiple of the relevant class size ($s_r = n$). We shall denote these Generality-Recall= n Precision Graphs as GR_nP Graphs; obviously the GR_iP Graph corresponds to the GR₁P Graph. By showing system performance for GR₁P and GR₂P indicating the performance for $s_r = 1$ and $s_r = 2$, the usability of the system for Total Recall would be well characterized. Its function can be compared to the Bull's Eye Performance (BEP) measure used in MPEG-7 for shape and motion descriptors⁵ but extended to include the effect of *generality* on the performance. Another well-known associated overall measure (but without taking generality into account) is van Rijsbergen's E-measure³ (for the connection between this measure and our approach see¹⁴).

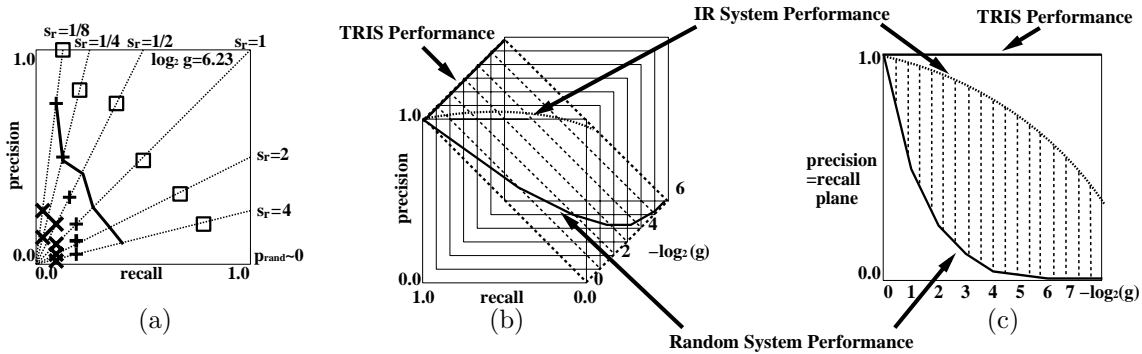


Figure 3. (a) Average p, r curve (connected line) from 3 individual retrieval tests (relevant class size=8) showing individual p, r results for different relevant scopes; (b) The 3-dimensional GReP Graph, a complete CBIR performance space with the $p = r$ plane (with random and $s = c$ results for different generality values) holding the GRiP Graph; (c) The 2 dimensional GRiP Graph: $p = r$ values for scope size=relevant class size as a logarithmic function of generality

4. EXPERIMENTAL RESULTS: PORTRAIT AND LOGO DATABASE

In our experiments (see also¹⁵), we have used a test database with only gray-level images and all from a narrow-domain. This means a sort of double handicap: we have to rely on intensity distribution features (shape and/or texture features) to cope with the fact that color/gray-level histogram features cannot distinguish well between large groups of gray-level images.

4.1. Defining a Class of Standard User Queries

One way of forming test queries is by collecting user queries and providing them with hand-checked image classes that serve as ground-truth during evaluation. The trouble with these queries, e.g. “find me all images that contain a table”, is that although it is not hard to decide image by image whether each image is in the table-class or not, it takes an enormous amount of time to build ground-truth for a large set of such queries. For our database, 21,094 portraits and 21,094 logos (at the back of the portraits), building up a statistically significant number of test queries with hand-checked ground-truth is not feasible.

Instead, we have implemented a ground-truth for two questions that can often be associated with a class of relevant images given different search images taken from the database itself. In these cases, the database can be seen as a multi-class division within an embedding of possibly non-class items; ground-truth can then be hand-checked for all classes while going through the database once. This set-up uses binary class labels: each image can be a member of one class at maximum.

4.2. The Making of Ground-Truth for the User Queries

The two problems we considered to generate queries are:

- Is there a (noisy) duplicate of this portrait?
- Are there other images with this (noisy) studio logo?

Part of this ground-truth exercise is described in.¹ Our database, the Leiden 19th-Century Portrait Database (LCPD) at <http://nies.liacs.nl:1860>, consists of Dutch studio portraiture, so called “Cartes de Visite” (CdV for short), that were produced between 1860 and 1914 by the millions. Customers were usually provided with a dozen copies of their portrait. At the back of the portraits, a studio logo is often present. Over the years different keeping conditions of the cartes distributed among relatives and friends gave rise to differences due to bleaching, staining, and/or annotation (names, dates, collection numbers) between original copy sets of portraits and/or logos. These effects were considered additive noise in our model of the database as a multi-class image collection. As a consequence none of the scanned images is a digital copy of any other image.

Since duplicates and logos mostly come from the same studio, a first division of the portrait database was made based on textual information about the studio, town, and address resulting in 3650 studio classes. After this step, the finding of duplicates and identical logo classes was restricted to detection of duplicates and grouping of different logos into clusters within each of the 3650 studio entries. That way 238 (noisy) duplicate pairs for the portraits and 1856 (noisy) logo classes with at 2 till 300 members (average size 8) could be formed efficiently by going through the image database a second time (studio by studio) for 42,188 images in all. This way 15,324 image queries with ground-truth answers have been generated.

4.3. Methods: Differences in Input Preparation, Indexes, and Retrieval

The following procedure describes how portraits were digitized and preprocessed before feature vectors were formed. The original portraits were scanned at 300 dpi and down sampled by averaging to create digital input at a range of resolutions; this resolution in dpi is one of the parameters of a CBIR method. Because most of the feature vectors we wanted to extract are sensitive to scale, rotation, and translation, all the digitized images were made invariant to these geometric changes by using a uniform resolution, standard orientation, and standard cropping procedure for all scanned images. All images were also made invariant to some of the lighting effects by contrast-stretching. The images were then ready for feature extraction based on the intensity-domain. For those feature vectors obtained from the gradient- or binarized gradient-domain, the Sobel 3x3 gradient magnitude image was formed and thresholded into binary images where needed.

For all methods compared here, images underwent the same input preparation phase; the main differences are the way feature vectors are formed during the index phase:

- RANDOM: input preparation, no feature vector, random ranking, standard scope
- LBP: Local Binary Pattern as defined in¹⁶; a pattern histogram with 256 entries
- LBPtG: variant of LBP with t=the threshold value applied to local gradient magnitude to determine whether the pattern contributes to the LBP histogram
- TRIGRAM512: as defined in¹⁷; a pattern histogram with 512 entries
- TRIGRAM510: variant of TRIGRAM512 by omitting the two homogeneous patterns (all black/all white)
- INTPROJ/GRADPROJ/BINGRADPROJ: Projections (horizontal and vertical) as defined in¹⁸ obtained from either the intensity-, the gradient- or the binarized gradient domain.
- BINtGRADPROJ: variant of BINGRADPROJ with t=the threshold value applied to gradient magnitude during binarization.

The resolution in dots per inch (dpi), at which feature vectors were formed, is added to the name of the method. The retrieval phase uses the same similarity measure (L_1) and class normalized standard scopes of $n \cdot \text{class-size}$. The optimization of similarity measures as such was part of earlier research reported in.¹⁹

4.4. Evaluation Measures: Recall, Precision, and Generality for Class Size Normalized Scopes

Like advocated in our evaluation paper,¹⁰ we have chosen an approach that normalizes performance with respect to the size of the class of relevant items and takes into account that by definition precision and recall are mutually dependent measures. With the normalization, $\text{scope} = n \cdot \text{class-size}$, precision and recall are connected by $\text{precision} = \text{recall} / n$.

As described earlier in this paper, instead of extending precision-recall graphs to a three-dimensional precision-recall-generality graph we prefer to use two-dimensional graphs at specific intersections of that three-dimensional performance graph. The first graph has the same form as the old precision-recall graph, but is restricted to a constant generality value and augmented by class size related integer scope lines. The second graph, at constant $\text{scope} = n \cdot \text{class-size}$, shows $\text{precision} = \text{recall} / n$ values as a function of generality. The generality value, equal to the expected random retrieval value $n \cdot \text{scope} / \text{database-size}$, is plotted as its negative \log_2 so that the generality range near nil is stretched in a compact way and indicates how, for each unit growth in generality, the performance changes with each successive doubling of the embedding.

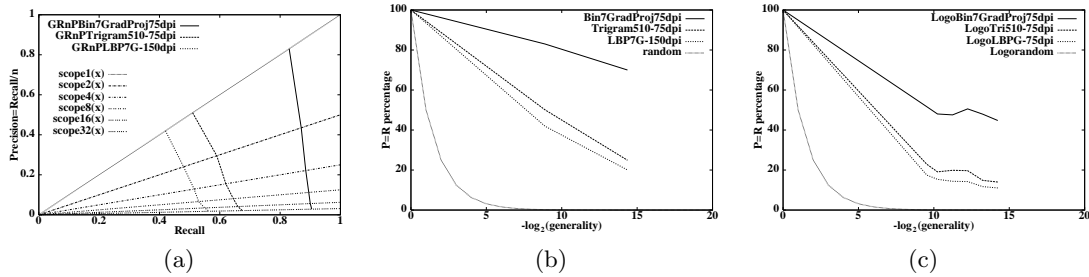


Figure 4. Indexing methods BINGRAD7PROJ, TRIGRAM510 and LBP7G compared to RANDOM retrieval Duplicate retrieval: precision=recall/n graphs at generality=0.02= random level, radiating lines for successive doubling of scope; (b) class size normalized recall=precision as a function of $\log_2(\text{generality})$; (c) same for Logo class retrieval

For evaluation purposes each member of a duplicate pair or logo class was taken in turn to automatically retrieve the remaining member(s). Measures were always averaged over all the class members and stored as class size normalized precision, recall, generality values for each CBIR method. Average performance figures were therefore obtained for the 238 duplicate and 1856 logo classes before being averaged over all duplicates or all logos with equal class sizes.

4.5. Optimizing Individual CBIR Methods

Numerous evaluation runs were made for each method to optimize parameter settings for that particular method. The following input parameters were evaluated: (1) Is there a best down sample resolution? The input resolution of 300dpi was varied by down sampling between 150 dpi and 15 dpi; measurements show that for most methods 75 dpi turned out to give optimal performance, although this effect is not large. As a result a first quick indexing run on resolutions as low as 15dpi can be done to downsize the task for subsequent detailed comparisons. (2) What is the optimal threshold for gradient magnitude binarization? We found that a threshold value around 7 (gradient values obtained from contrast stretched intensity images) is optimal; this value corresponds to a few times the average noise level.

Suggested improvements of individual methods like dropping those patterns from LBP when the local region is almost uniform (noise patterns are to be expected then), could in this set-up be evaluated and validated quickly: counting LBP patterns only when their formation region contains high-enough gradients (threshold of 7 in gradient-magnitude proved optimal) boosted the performance of this LBP variant, that we baptized LBP7G, by a factor of 2.

Based on earlier experiments with Trigrams reported in,²⁰ instead of using the full uniformly weighted trigram histogram of pattern counts (TRIGRAM512), we have used the better performing version where homogeneous black/white regions, on average 60 percent of the patterns, are not counted reducing the effective length from 512 to 510 (TRIGRAM510).

This evaluation of optimising the parameters of individual methods already convinced us that it is profitable to exclude homogeneous regions with only noise from contributing to the feature formation.

4.6. Comparing Optimized CBIR Methods

We present average performances (over 238 duplicate pairs) as intersections of the 3D performance graph in a constant Generality plane in Figure 4 (a) (the well-known but augmented precision-recall graph) and in the class-size normalized scope plane (precision=recall) in Figure 4 (b).

These comparisons clearly show the “generality effect” (performance diminishes within a growing embedding) and it is clear that all non-random indexing methods give a higher performance, and that, as an intensity distribution descriptor, projections (most successful variant BIN7GRADPROJ) highly outperform feature vectors based on the most successful variants of either LBP and Trigrams.

The results in Figure 4 (a) show that for the best performing method BIN7GRADPROJ, not much is gained by using scope values > class size: about 5% better results are obtained by doubling the scope to 2*class-size;

about 10% better results are obtained by taking a 32-fold scope size ($32 \cdot \text{class-size}$). For the worst performing method LBP7G the gain is greater: about 15% better results are obtained by doubling the scope to $2 \cdot \text{class-size}$; about 60% better results are obtained by taking a 32-fold scope size ($32 \cdot \text{class-size}$). Less performing methods gain more by increasing normalized scope than high performing ones.

The dependency of performance on the level of generality is also clearly visible. Figure 4(b) shows a steady performance decline for duplicate retrieval at each successive doubling of the irrelevant embedding (at increments of 1 in the value of $-\log_2(\text{generality})$). The dependence of performance on the level of generality is less clearly visible for the averages of logo classes, shown in Figure 4(c), with class sizes varying from 2 till 32 covering a range of $-\log_2(\text{generality})$ of almost 5. This graph only shows performance figures for the most successful methods within the maximum embedding available.

5. LABORATORY SYSTEMS VERSUS PRACTICAL SYSTEMS

We have shown that for a complete performance evaluation, one has to carry out controlled retrieval tests, with queries for which ground-truth can provide the relevant class sizes. The performance is measured for various ranking methods, within a range of scope and generality values.

Since it is often too costly and labor intensive to construct the complete ground-truth for the queries used, we will indicate what could be done in terms of evaluation when relevant class sizes, and as a result *recall* and *generality* values, are missing.

First, let us make a distinction between Laboratory CBIR systems and Practical CBIR systems. We propose to reserve the name Laboratory CBIR system for those performance studies where complete ground-truth has become available. For these systems a complete performance evaluation, in the form of Generality-Recall-Precision Graphs, for a set of test queries and for a number of competing ranking methods is provided.

Any CBIR retrieval study that lacks complete ground-truth will be called a Practical system study. In Practical system evaluation, one normally has a set of queries and a database of known size d . Because ground-truth is missing, relevant class size c is unknown. The only two free controls of the experimenters are scope s and ranking method m . Relevance judgements have to be given, for the retrieved items within the scopes used to determine the number of relevant answers v within the scope s . Of the three Laboratory system evaluation parameters *precision*, *recall*, and *generality* only *precision* $= v/s$ is accurately known. For *recall* due to knowing v but not c only a lower bound $v/(d - s + v)$ is known. For generality only a lower bound $g = v/d$ is known. In general, for practical studies, one characterizes the performance as Precision-Scope Graphs or one uses single measures obtained from the weighted ranks of the relevant items within scope.

The problem with any Practical system study result is that one cannot interpret the results in terms of “expected completeness” (*recall*), and the results are therefore only useful in terms of economic value of the system. How many items will I have to inspect extra to obtain an additional relevant item? How much time will it cost me to retrieve a certain number of relevant results?

Actually, with some extra effort the analysis of a Practical system can be enhanced to that of an estimated Laboratory system, by using the fact that *generality* in terms of relevant fraction is identical to the expected *precision* (see Eqn 12) when using a random ranking method.

Experimenters that have access to the ranking mechanism of a retrieval system can thus obtain estimates for generality g , and hence relevant class size c and recall r to complete their performance evaluation. The extra effort required would be the making of relevance judgements for a series of randomly ranked items within some long enough scope.

6. FINAL REMARKS

The addition of the third normalized IR system parameter, *generality*, in performance studies makes that single number characterizations like van Rijsbergen’s *E*-measure have become a function of generality. In general, for a small embedding the *precision*, *recall* curves will be close to the ideal curve, with a corresponding high valued *E*-measure. For a growing embedding size the *p*, *r* curves will gradually turn into the well-known hyperbolic form of very large databases with an associated low valued *E*-measure. So for large enough databases, the effectiveness

may well drop to a value no longer acceptable for users, unless the fall in effectiveness is encountered by a rise in feature vector effectiveness (like a better or longer feature vector per item). The fall in effectiveness of the Alta-Vista text search engine on Internet for instance is momentarily successfully repaired, by the approach followed by Google, that uses additional features (in this case the recursive structure of links on the Internet to the retrieved items for a weighted sorting of the results) to regain an acceptable effectiveness at the top of the result list. This process of slow degradation, caused by the continuous growth of the database, and sudden improvement gains, due to more successful arrangements of indexing and clustering methods, will probably be seen and be needed more often in the years to come.

6.1. Wide- or Narrow-Domain?

Although we have shown in this paper that all the relevant parameters of IR systems can be brought to light with additions to the PR graph, there remains a problem with generality for IR performance tests. How the performance degrades within larger embeddings, greatly depends upon how similar or dissimilar the embedding items are, with respect to the relevant items, and how compact the clusters of relevant items are. In this light, one often makes the distinction of wide-domain versus narrow-domain databases. A wide domain database has a more uniformly filled feature space and since feature space in typical IR systems is high-dimensional, performance will remain quite high, even for a very large embedding (millions of items) especially when the clusters are quite compact. A narrow-domain database, on the other end of the spectrum, is characterized by embedding items that are highly similar, and therefore IR performance may degrade much more rapidly. Most commercial applications of content-based image or sound retrieval (with a notable exception for Internet search engines) are more likely to be narrow- than wide-domain databases; it is therefore crucial to develop features that remain distinctive in large narrow-domain systems. Displaying generality information is therefore only part of the solution in comparing different IR systems: even equal generality level systems may be difficult to compare, if one of them is a wide-domain system (like an Internet image search engine), and the other one is a narrow-domain system (like a world-wide company logo search system).

Internationally agreed upon test sets and fixed test databases, for both wide and narrow domain databases, are needed to solve the problem of finding the better ranking methods for a large set of CBIR applications. We therefore welcome efforts like Benchathlon, a benchmark for CBIR.

6.2. Discussion

We surveyed Content-Based Image Retrieval performance graphs and found them to be lacking generality information. We also noted that one is not aware of the scope information present in a Precision-Recall Graph and the lack of comparison with random performance. As a result, commonly used Precision-Recall Graphs can only be used when plotting *precision, recall* values obtained under a common, mentioned, *generality* value. This generality value coincides with the random performance level. Due to the dependency of precision and recall, their combined values can only lay on a line in the PR Graph determined by the scope used to obtain their values. Scopes therefore can be shown in the PR Graph as a set of radiating lines. A normalized view on scope, relevant scope, makes the intuitive notion of scope much more simple.

As a result average *precision, recall* values, obtained by either averaging precision values in recall boxes or vice versa, conflict with the way precision and recall are dependently defined. Averaging *precision, recall* values should be done along constant scope lines, and only for those *p, r* values that have the same generality value. During performance measurements *p, r, s_r, g* values should be collected and averaging should be restricted to those measurements that share common *s_r, g* values.

To complete performance space we extended the traditional two-dimensional Precision-Recall graph to the three-dimensional GReP Graph (Generality-Recall-Precision Graph) by adding a logarithmic generality dimension. This performance space is a continuous series of PR Graphs.

For Total Recall System performance, we advocated a comparison with the Total Recall Ideal System performance, by using a special plane in the GReP Graph, the two-dimensional GRiP Graph (Generality-Recall=Precision Graph) showing the diagonal of the PR Graphs (Recall and Precision being equal on the diagonal of the PR graph) as a logarithmic function of generality. This gives a much more true evaluation of the performance degradation as a function of generality. In this way, statements can be made about what to

expect from a scaled-up system given the performance degradation of the laboratory system within the generality range tested. We also make a distinction between Laboratory CBIR Systems and Practical CBIR Systems: for Laboratory Systems complete ground-truth is available, for Practical Systems it is lacking but can be estimated.

The extensions to performance graphs, suggested in this paper make it possible to better compare performance figures within growing CBIR Systems (by explicit mentioning of the generality level), and make it possible to infer precision and recall as a function of normalized scope, reducing the need for additional *precision* or *recall* versus *scope* graphs next to Precision-Recall graphs. For the evaluation of the performance, in relation to continually growing database sizes, the GRiP Graph (Generality-Recall=Precision Graph) offers the best overall IR System performance overview, since this graph shows how well the system in question approaches TRIS (the Total Recall Ideal System).

REFERENCES

1. D. Huijsmans, N. Sebe, and M. Lew, "A ground-truth training set for hierarchical clustering in content-based image retrieval," in *Visual 2000*, pp. 500–510, 2000.
2. G. Salton, *The SMART retrieval system*, Prentice Hall, 1971.
3. C. van Rijsbergen, *Information Retrieval*, Butterworths, London, 1979.
4. E. M. Voorhees and D. Harman, *Proc. TREC*, 1999.
5. MPEG-7 *Special Issue of IEEE Trans. Circuits and Systems for Video Technology* **11**(6), 2001.
6. K. Porkaew, K. Chakrabarti, and S. Mehrotra, "Query refinement for multimedia similarity retrieval in MARS," in *ACM Multimedia*, pp. 235–238, 1999.
7. N. Vasconcelos and A. Lippman, "A probabilistic architecture for content-based image retrieval," in *CVPR*, pp. 216–221, 2000.
8. C. Baumgarten, "A probabilistic solution to the selection and fusion problem in distributed information retrieval," in *SIGIR*, pp. 246–253, 1999.
9. H. Müller, W. Müller, D. M. Squire, S. Marchand-Maillet, and T. Pun, "Performance evaluation in content-base image retrieval: Overview and proposals," *Pattern Recog. Letters* **22**, pp. 593–601, 2001.
10. D. Huijsmans and N. Sebe, "Extended performance graphs for cluster retrieval," in *CVPR*, pp. 26–31, 2001.
11. D. Gokhale and S. Kullback, *The Information in Contingency Tables*, M. Dekker, 1978.
12. J. Tague-Sutcliffe, "The pragmatics of information retrieval experimentation, revisited," *Information Processing and Management* **28**(4), pp. 467–490, 1992.
13. V. Raghavan, P. Bollmann, and G. Jung, "A critical investigation of recall and precision as measures of retrieval system performance," *ACM Trans. Information Systems* **7**, pp. 205–229, 1989.
14. D. P. Huijsman and N. Sebe, "How to complete performance graphs in content-based image retrieval: Add generality and normalize scope," *IEEE Trans. on PAMI*, to appear, 2004.
15. D. Huijsmans and N. Sebe, "Content-based indexing performance: A class size normalized precision, recall, generality evaluation," in *ICIP*, pp. 733–736, 2003.
16. T. Ojala, M. Pietikainen, and D. Harwood, "A comparative study of texture measures with classification based on feature distributions," *Pattern Recognition* **29**(1), pp. 51–59, 1996.
17. D. Huijsmans, S. Poles, and M. Lew, "2d pixel trigrams for content-based image retrieval," in *Proc. Int. Workshop on IDB-MMS*, pp. 134–145, 1996.
18. D. P. Huijsmans and M. S. Lew, "Efficient content-based image retrieval in digital picture collections using projections: (near)copy location," in *ICPR*, **3**, pp. 104–108, 1996.
19. N. Sebe, M. Lew, and D. Huijsmans, "Towards improved ranking metrics," *IEEE Trans. PAMI* **22**(10), pp. 1132–1143, 2000.
20. D. Huijsmans and M. Lew, "Quality measures for interactive image retrieval with an evaluation of two texel-based methods," in *ICIAP*, pp. 22–29, 1997.