

CONTENT-BASED INDEXING PERFORMANCE: A CLASS SIZE NORMALIZED PRECISION, RECALL, GENERALITY EVALUATION

Dionysius P. Huijsmans

Leiden Institute of Advanced Computer Science
Niels Bohrweg 1, 2333 CA Leiden
The Netherlands
email: huijsman@liacs.nl

Nicu Sebe

Faculty of Science, University of Amsterdam
Kruislaan 403, 1098 SJ Amsterdam
The Netherlands
email: nicu@liacs.nl

ABSTRACT

Progress in Content-based Image Retrieval (CBIR) is hampered by the lack of good evaluation practice and test-benches. In this paper, we raise the awareness of all the parameters that define a content-based indexing and retrieval method. Extensive ground-truth, 15,324 hand-checked image queries, was developed for a portrait database of gray-level images and their backside studio logo's. Our aim was to clearly demonstrate the diminishing effect of a growing embedding on performance figures, and the establishment of a reliable ranking of several suggested CBIR gray-level indexing methods. This evaluation scheme was used first to optimize a number of parameters defining the detailed workings of each method. The database, standard image queries, ground-truth, and evaluation scripts are offered for inclusion in an evaluation site like Benchathlon.

1. INGREDIENTS FOR A STATISTICALLY MEANINGFUL CBIR TEST SET-UP

In [5], we have stated what should be added to recent CBIR system evaluation proposals like [6] to set up an evaluation procedure that completely describes the influence of the relevant parameters of the indexing and retrieval methods.

Our main objection, to the so far standard practice of using precision-recall graphs and precision-scope graphs to evaluate and compare methods, was that the influence of the relative size of the relevant image class versus the size of the irrelevant (embedding) items in a database, characterized as generality, has somehow got lost after the first years of text-retrieval evaluations as is shown by [9] and [8]. Therefore, our first objective with this paper is to show convincingly (statistically speaking) that a growing embedding, of irrelevant items around relevant image classes, diminishes the retrieval performance. This effect characterized by generality (class-size of relevant items/ size of testdatabase) should be combined with the well-known performance measures: precision (retrieved number of relevant items within the scope/scope size) and recall (retrieved number of relevant items within the scope/class size of relevant items).

A second objection against present presentations of performance figures is the fact that retrieval scopes are not normalized with respect to the size of the relevant image class.

The effect this has on the test set-up and normalization was illustrated in our performance evaluation paper [5].

A major obstacle toward the selection of promising methods for inclusion in commercial image retrieval systems, is the slow development of image search test benches. Initiatives like Benchathlon (<http://www.benchathlon.net>) deserve more than a one-time backing. Since all indexing and retrieval methods will perform differently for each different set of user queries carried out within each specific embedding, we have to develop a mass of standard user queries that are used within well-defined embeddings to single out those indexing and retrieval methods that can be expected to perform better in general.

1.1. CBIR method definition: essential parameters?

A CBIR method can be thought of as a successful combination of indexing and ranking techniques. From a narrow-minded perspective this amounts to the extraction of a feature vector (index) and the sorting of images on the basis of similarity measured with respect to a search image (the retrieval); moreover a scope is used to limit the display of retrieved items for visual inspection by a user.

In our view different indexing methods may involve much more than merely taking a different recipe for calculating a feature vector. It includes the precise description of all digitization and preprocessing steps taken before digital images are transformed into feature vectors. These steps should be considered part of the indexing phase of a CBIR method and all relevant parameters (e.g. threshold settings) applied in sub-processes should be described and quantified.

Many more parameters than a fixed similarity measure and a scope on the ranking list may be considered as essential parts of the CBIR retrieval method. Multi-dimensional similarity measures, the difference measure itself (for an overview of optimal one-dimensional metrics see [10]), the effect of weighting feature vector elements differently (for instance due to relevance feedback), and the subsequent clustering methods to reorganize and diminish the number and sorting of retrieved items shown, determine which resulting images are shown to the user. Our evaluation procedure must make all these influences visible because otherwise differences in performance might be attributed to

the wrong causes, and progress toward better CBIR performance in general would become largely erratic.

2. DATABASE: RELEVANT IMAGES EMBEDDED IN IRRELEVANT IMAGES

Any indexing and retrieval method may give reasonable results when its feature space is sparsely filled (due to the high dimensionality of the feature vector and/or due to a small or diverse embedding). Gray value histograms for instance, quickly drop in performance within a growing embedding, whereas color histograms that span a higher dimensional feature space are much more resistant to a growing embedding especially when the embedding items are very diverse. In the long run however, even in wide-domain embeddings, like all images on Internet, color-histogram features will fail to distinguish between too many images.

2.1. A narrow-domain gray-level image database

In our experiments, 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.

2.2. 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 as studio prop", 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, 21,000 portraits and 21,000 logo's (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 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.

2.3. 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 [4]. 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, that were produced between 1860 and 1914 by the millions. Costumers 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 logo's. 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 logo's 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 is restricted to detection of duplicates and grouping of different logo's 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 til 300 members (average size 8) could be formed efficiently by going through the image database a second time (studio by studio) for 42,000 images in all. This way 15,324 image queries with ground-truth answers have been generated.

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 in [7]; a pattern histogram with 256 entries
- LBPG: variant of LBP with t = threshold value local gradient magnitude to determine whether pattern contributes to the LBP histogram
- TRIGRAM512: as defined in [1]; 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 in [2] obtained from either the intensity-, the gradient- or the binarized gradient domain.
- BINtGRADPROJ: variant of BINGRADPROJ with t = threshold value 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 *class-size.

3.1. Evaluation measures: Recall, precision, and generality for class size normalized scopes

Like advocated in our evaluation paper [5], 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, $\text{scope}=n$ *class-size, precision and recall are connected by $\text{precision}=\text{recall}/n$.

As described in [5], 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 3D 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 $\text{scope}=n$ *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 *scope/database-size, is plotted as its negative $2\log$ 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.

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 logo's with equal class sizes.

3.2. Optimizing individual CBIR methods

Numerous evaluation runs were made for each method to optimize parameter settings for that particular method.

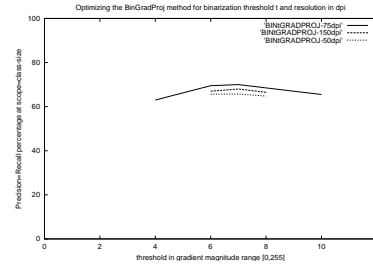


Fig. 1. Plot of scope normalized recall=precision values versus gradient magnitude threshold for various down sampling resolutions 75, 150 and 50 dpi (top til bottom curve)

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 can be seen from Figure 1. 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 find, as can be seen in Figure 1, 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 LBPG, by a factor of 2.

Based on earlier experiments with Trigrams reported in [3], 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 method by method evaluation already convinced us that it is profitable to exclude homogeneous regions with only noise from contributing to the feature formation.

3.3. 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 2 (a) (the well-known but augmented precision-recall graph) and in the class-size normalized scope plane ($\text{precision}=\text{recall}$) in Figure 2 (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

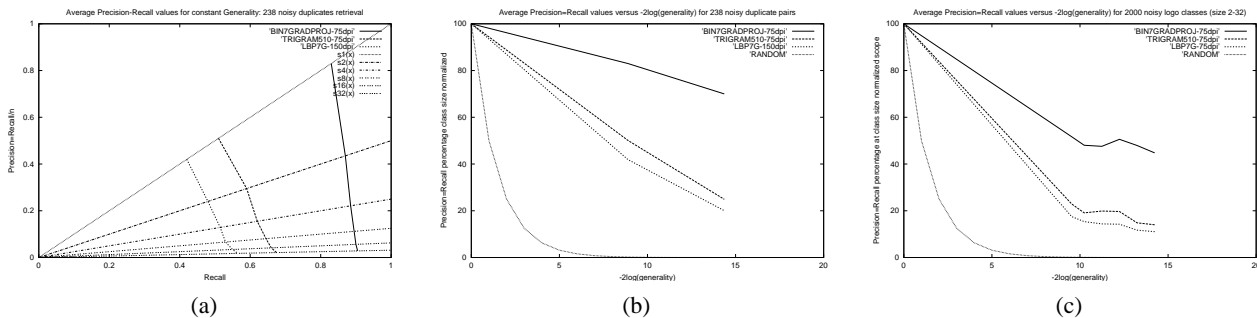


Fig. 2. Indexing methods BIN7GRADPROJ, TRIGRAM510 and LBP7G compared with RANDOM retrieval (a) Duplicate retrieval: precision=recall/n graph at generality=0.02=random level with scope size 1,2,4,8,16 and 32; lower radiating lines $p=r/n$ for bigger scopes; (b) Duplicate retrieval: class size normalized recall=precision as a function of $\log(\text{generality})$; (c) Same for Logo class retrieval

distribution descriptor, projections (most successful variant BIN7GRADPROJ) highly outperform feature vectors based on either LBP and Trigrams and their more successful variants.

The results in Figure 2(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 $2 \cdot \text{class-size}$; 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 2(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 $-2\log(\text{Generality})$).

The dependence of performance on the level of generality is less clearly visible for the averages of logo classes, shown in Figure 2(c), with class sizes varying from 2 till 32 covering a range of $-2\log(\text{generalities})$ of almost 5. This graph only shows performance figures for the most successful methods within the maximum embedding available.

4. CONCLUSIONS

The performance figures obtained from these statistically reliable class size retrieval rates confirm the statement that generality as a performance characteristic should be taken into account: performance figures diminish within a relatively greater embedding of irrelevant items. With the input preparation phase and ranking measure used, the performance differences show that the CBIR indexing methods presented, for gray-level image data, can be ranked as follows:

RANDOM < LBP7G < TRIGRAM510 < BIN7GRADPROJ

We offer our database plus ground-truth testing environment for inclusion in Benchathlon. For the time being, our 15,324

standard image queries with ground-truth database, can be used to test any suggested CBIR method. Until Benchathlon is an integrated test bed, we are willing to do testing on demand (when given enough details to implement a suggested index and retrieval scheme within our test database). We also determined error-bars to the averages shown above. They turn out to be quite large, even for these relatively simple classes of user queries.

5. REFERENCES

- [1] D.P. Huijsmans, S. Poles, M.S. Lew, 2D pixel trigrams for content-based image retrieval. in Smeulders A., Jain R. (eds), Image databases and Multi-Media search, Proc. 1th Int workshop IDB-MMS, 139-145, 1996.
- [2] D.P. Huijsmans, M.S. Lew, Efficient Content-Based Image Retrieval in Digital Picture Collections using Projections: (near)Copy Location, ICPR'96, Vol 3, 104-108, 1996.
- [3] D.P. Huijsmans, M.S. Lew, Quality Measures for Interactive Image Retrieval with an Evaluation of two Texel-Based Methods, ICIAP'97, LNCS 1311, 22-29, 1997.
- [4] D.P.Huijsmans, N. Sebe, M.S. Lew, A Ground-Truth Training Set for Hierarchical Clustering in Content-based Image Retrieval, Visual 2000, LNCS 1929, 500-510, 2000.
- [5] D.P. Huijsmans, N. Sebe, Extended Performance Graphs for Cluster Retrieval. CVPR 2001, 126-31.
- [6] H. Muller, W. Muller, D. McG. Squire, S. Marchand-Maillet, and T. Pun, Performance evaluation in content-base image retrieval: Overview and proposals, Pattern Recog. Letters, Vol 22, 593-601, 2001.
- [7] Ojala T., Pietikainen M and Harwood D., A comparative study of texture measures with classification based on feature distributions. Pattern Recognition, vol 29-1, pp 51-601, 1996 .
- [8] C. J. van Rijsbergen, Information Retrieval (second edition), Butterworths, London, 1979.
- [9] G. Salton, The "Generality" Effect and the Retrieval Evaluation for Large Collections, Journal American Society for Information Science, Jan-Feb 1972, 11-22.
- [10] N. Sebe, M.S. Lew, D.P. Huijsmans, Towards Improved Ranking Metrics, IEEE Trans. PAMI, Vol 22, No. 10, 1132-1143, 2000.