Improving the Boosted Correlogram

Nicholas R. Howe and Amanda Ricketson

Smith College, Northampton, MA, USA,
nhowe@cs.smith.edu

Abstract. Introduced seven years ago, the correlogram is a simple statistical image descriptor that nevertheless performs strongly on image retrieval tasks. As a result it has found wide use as a component inside larger systems for content-based image and video retrieval. Yet few studies have examined potential variants of the correlogram or compared their performance to the original. This paper presents systematic experiments on the correlogram and several variants under different conditions, showing that the results may vary significantly depending on both the variant chosen and its mode of application. As expected, the experimental setup combining correlogram variants with boosting shows the best results of those tested. Under these prime conditions, a novel variant of the correlogram shows a higher average precision for many image categories than the form commonly used.

1 Introduction

An image rarely reveals anything of interest in its raw pixel color data. For most tasks, pertinent information must be extracted computationally from the raw pixel intensities, yielding new forms of data that describe the image more effectively for the task at hand. Both image retrieval and the related task of image classification depend on effective image descriptors for success. Yet the development of effective descriptors for image and video indexing remains an area of basic research. Although not suitable for all tasks, simple descriptors that represent an image holistically (rather than by parts or regions) have proven remarkably effective in many areas, and are widely used, both outright for indexing and as components in larger systems. Six or seven years ago, the holistic descriptor of choice was the color histogram; today, as judged by recent citations, it is the color correlogram [2, 11].

Given the success of the color correlogram as an image descriptor for indexing and classification, it is somewhat surprising how little research explores the details of its implementation and possible variants. In part this may be attributed to a sentiment among researchers that holistic representations lack the sophistication required for "real" image retrieval. Some denigrate the correlogram as too simple to capture the nuances of real semantic categories. Yet in experiments it handily beats other supposedly more nuanced representations [6, 8]. More to the point, the fact of its widespread use merits a second look. While the correlogram's holistic approach may not be in tune with current thinking about how

image retrieval should work, it offers great strengths as a *component* of a larger system. This observation motivates the work in this paper, which seeks ways to improve upon the correlogram in certain applications.

The next section of the paper considers the origins and definition of the standard correlogram, and proposes several variants for investigation. A short summary of recent work in boosting for classification and retrieval follows. Section 3 describes a set of experiments comparing the correlogram variants on a selection of image classification/retrieval tasks. Finally, Section 4 concludes with an analysis of the lessons learned and potential further steps.

2 Correlogram Variants and Boosting

The color correlogram has proven its worth as an image descriptor for both comparison and retrieval. Relatively compact and simple to implement, yet more subtle and powerful than the color histogram, it has become perhaps the most widely used image descriptor today. Previous work has shown that applying boosting techniques to the correlogram representation yields a high quality image classifier, better than many other published boosted image classification/retrieval algorithms [7], and that boosting can function as a feature selector [1, 14].

The descriptor that has become known as the correlogram comprises a feature vector computed on an image discretized into n color bins. ($n = 128$ in this paper.) Each component has a succinct probabilistic interpretation: given a pixel p of color c_i, what is the chance that a pixel chosen at random from a specified neighborhood around p also has color c_i? The standard treatment uses concentric ring neighborhoods with square radii of 1, 3, 5, and 7 pixels, allowing for fast computation via dynamic programming. In the equations below, $\Phi(p)$ represents the color of pixel p, and $d(p_1, p_2)$ represents the chessboard distance between pixels p_1 and p_2.

$$C_{c_i,r_1,r_2} = P\left(\Phi(p_2) = c_i \,|\, \Phi(p_1) = c_i \wedge p_2 \in B_{r_1,r_2}(p_1)\right) \tag{1}$$

$$B_{r_1,r_2}(p_1) = \{p_2 \,|\, r_1 < d(p_1, p_2) \leq r_2\} \tag{2}$$

The correlogram as described above first appeared in 1997 [10] and was developed further as part of a family of related descriptors in the Ph.D. dissertation of Jing Huang [9]. Huang referred the commonly used descriptor given above as the *banded autocorrelogram*. In this terminology, *banded* refers to the square ring neighborhoods used to compute the correlogram, and the *auto-* prefix indicates that all the measurements involve frequencies of pixels of the same color. Huang describes but does not further explore a more general set of statistics defined over a set of distance bands and all possible pairs of colors (c_i, c_j). A single component of this descriptor considers all pixels of some color c_i, and measures the fraction of pixels within a particular distance band that are a second color c_j.

$$C^*_{c_i,c_j,r_1,r_2} = P\left(\Phi(p_2) = c_j \,|\, \Phi(p_1) = c_i \wedge p_2 \in B_{r_1,r_2}(p_1)\right) \tag{3}$$

Although the general correlogram requires significantly greater storage than the autocorrelogram, two considerations argue against writing it off immediately. First, recent research on other large image descriptors has shown that they can be effective if applied in combination with effective feature selection algorithms [14]. Second, study of the general correlogram may motivate more compact representations that nevertheless capture the additional information contained in the general correlogram.

This paper introduces a novel image descriptor that represents a compromise in both size and descriptiveness between the autocorrelogram and the general correlogram. Called the *color band correlogram*, it groups colors into color distance bands analogous to the spatial distance bands of the standard correlogram. Each component of the color band correlogram corresponds to a specified initial color c_i, a distance band specified by the bounds r_1 and r_2, and a color band specified by perceptual difference in color space from c_i lying between ρ_1 and ρ_2. The value of the component equals the mean fraction of pixels falling within the specified spatial neighborhood that have colors in the specified color band.

$$C^{CB}_{c_i,r_1,r_2,\rho_1,\rho_2} = P\left(\Phi(p_2) \in \beta_{\rho_1,\rho_2}(c_i) \,|\, \Phi(p_1) = c_i \wedge p_2 \in B_{r_1,r_2}(p_1)\right) \quad (4)$$

$$\beta_{\rho_1,\rho_2}(c_i) = \{c_j \,|\, \rho_1 < \delta(c_i, c_j) \leq \rho_2\} \quad (5)$$

In the equation above, δ represents a perceptual distance function in color space, and ρ_1 and ρ_2 are similarity bounds demarking a set of colors around the central color c_i. In practice correlograms may be computed for two or three color bands, corresponding respectively to an exact color match c_i, a close color match (a handful of colors directly surrounding c_i), and perhaps a more relaxed color match (colors similar to c_i but not in the closely matching category). With three color bands, the color band correlogram requires three times the storage of the autocorrelogram. This reprises the difference in storage between the histogram and the autocorrelogram, which differs by a factor equal to the number of distance bands.

The extra information in the correlogram variants described above may allow higher accuracy in some cases, but may also prove a liability if the inclusion of less relevant features drowns out the more important ones. In other words, the compactness and simplicity of the autocorrelogram may be an advantage under some circumstances. Interestingly, others have studied image descriptors that include large numbers of mostly irrelevant features. Although these descriptors yield poor results when used directly for retrieval, they can become competitive when applied in conjunction with a feature selection algorithm [14]. Boosting has served successfully in this capacity, although it was not originally designed as a feature selector.

The experiments in Section 3 compare the performance of the three correlogram variants in both their original form and using AdaBoost [4] as a feature selector. We hypothesize that the correlogram variants that contain more information will benefit most from boosting, since the boosting process can act as a feature selector. With images where the extra information is relevant to the

query task, the more complex variants should outperform the autocorrelogram; where it is not relevant they should do about the same. The unboosted variants, on the other hand, should suffer somewhat when they include extra features not relevant to the image category being retrieved. One caveat applies: if the amount of training data is not sufficient, boosting may not be able to properly extract features that generalize to unseen images. The experimental results should indicate whether this is a common problem in practice.

This paper breaks no new ground with regard to boosting algorithms themselves; the reader should refer elsewhere for details [5]. Boosting works by repeatedly learning to classify a labeled training set under different weightings of the training instances. The reweighting serves to focus effort on boundaries and special cases, sharpening the definition of the target class. Both theory and practice indicate that the weighted vote of all the classifiers created during training will be more accurate than the original unboosted classifier [12, 13].

Note that boosting is typically used not for retrieval but for *classification*, and it requires a training set of both positive and negative instances of the class to be retrieved. Yet it also can perform retrieval. Once trained, a boosted classifier assigns a score to any image that can be used for ranking of unknown images with respect to the trained category. Although some have developed ways to apply boosting within the canonical single-image query model [14], using it most naturally motivates a shift in methodology away from query-by-example toward query-by-category. For example, boosting could be used to train a library of classification models for keyword-based queries, or as input to some larger system. This paper adopts a methodology based upon trained image classifiers throughout, even for the unboosted experiments.

3 Experiments

The experiments divide naturally into two parts: those involving unboosted techniques, and those that involve boosted techniques. The methodologies are similar. All experiments share a 5x2-fold cross validation setup, a common classification testing framework [3]. They differ in the amount of training data used: the unboosted techniques can use all the available data, while the boosted experiments must hold some out (as described below).

For the unboosted descriptors, there are two further divisions into sets of experiments, depending upon the style in which the training data are used. The first style mimics query-by-example: each positive image in the training set forms a single-image query against the which images from the test set are ranked. The average of all these single-image queries gives the overall recall-precision figures for the test fold.

The second style of unboosted experiment builds an unboosted nearest-neighbor classifier. It selects the best exemplars of the class using a greedy additive approach: single images are added from the target class to the exemplar set one by one. The classification rate on the training set forms the criterion for selecting the next exemplar to add; when no new images can improve the train-

ing error, selection stops. The exemplar set then forms the positive examples for the nearest-neighbor classifier. Previous work has shown that this approach works better than simply using all the positive training instance for classification, since some of these may be particularly poor exemplars that can lead the classifier astray [8].

For the boosted experiments, the training data are further split into two equal subsets, one of which is used to train the boosted classifier, while the other (called the *holdout set*) is used to prevent overtraining. (Overtraining refers to situations where a classifier becomes too attuned to the particular set of examples used in training, and cannot generalize to the differences present in new data.) When performance on the holdout set ceases to improve, training stops. Although this method avoids overtraining, overall performance can be lower than if all the data were used for training. Nevertheless, the holdout set method maximizes fairness to the different methods, since they all receive optimal training on the data available.

The image library consists of 20100 images from the Corel photo CD collection, and is described in detail elsewhere [8]. Fifteen image categories chosen to represent a range of difficulty and subject matter make up the target classes. The names of the fifteen categories appear in the tables of results.

Tables 1 and 2 summarize the results of testing on the retrieval performance of the unboosted image descriptors. All numbers given in the tables are average precision. Table 1 shows the results for single-image queries, while Table 2 shows the results for the greedy-exemplar approach. Each row contains results for one image class, while the columns represent the autocorrelogram, two forms of color band correlogram, and general correlogram respectively. (The color band correlograms differ in that the first uses two bands, while the second uses three.) Since the random fold choice over five replications of the experiment leads to substantial variance, the standard deviation of each number shown in the table does not reliably indicate the significance of differences when comparing results between columns. A paired sample t-test accounts for the variance due to the random fold choice and reliably indicates which differences achieve statistical significance. The table uses bold type for performances of the correlogram variants that differ significantly from that of the autocorrelogram, and underlines the cases that represent improvements.

The two tables show that increasing the number of features without boosting tends to decrease the average precision. Although the color band correlograms do better on a few categories, the general correlogram (with the largest number of features by far) does uniformly worse than the autocorrelogram. These results suggest that irrelevant information in the additional features added to the correlogram variants contain is misguiding the retrieval process.

By contrast, boosting changes the results entirely. Table 3 summarizes the the retrieval performance of the boosted image descriptors, in the same format as the tables above. With boosting, the virtues of the correlogram variants become evident: the descriptors with the most features do the best. Although the large variances on some categories limit the number of statistically significant results

Table 1. Average precision for correlogram descriptors on 15 image classes, using unboosted single-image queries. From left to right, columns show the autocorrelogram, color band correlogram with two bands, color band correlogram with three bands, and general correlogram. Numbers that differ significantly from the autocorrelogram are bold, and improvements are underlined. Units are percentages; i.e., perfect retrieval = 100.

Class	Auto.	CB2	CB3	GC
Race Cars	3.4 ± 0.3	**2.6 ± 0.3**	**2.7 ± 0.3**	**1.0 ± 0.2**
Wolves	2.7 ± 0.2	**2.1 ± 0.2**	**2.2 ± 0.3**	**2.1 ± 0.3**
Churches	1.2 ± 0.1	**0.93 ± 0.08**	**0.94 ± 0.07**	**0.87 ± 0.15**
Tigers	10 ± 1	**8.3 ± 1.0**	10 ± 2	**8.4 ± 1.9**
Caves	1.9 ± 0.1	**1.4 ± 0.1**	**1.6 ± 0.1**	**1.3 ± 0.1**
Doors	1.4 ± 0.2	1.4 ± 0.2	**1.5 ± 0.3**	**0.96 ± 0.23**
Stained Glass	29 ± 4	**27 ± 3**	**32 ± 3**	**11 ± 2**
Candy	2.6 ± 0.4	**2.2 ± 0.3**	**2.1 ± 0.4**	**1.6 ± 0.3**
MVs	1.3 ± 0.2	1.3 ± 0.2	**1.2 ± 0.2**	**1.0 ± 0.2**
Bridges	1.2 ± 0.1	**0.99 ± 0.06**	**0.98 ± 0.05**	**1.0 ± 0.1**
Swimmers	4.2 ± 0.4	**5.3 ± 0.4**	**4.7 ± 0.3**	**1.4 ± 0.2**
Divers	12 ± 1	**4.7 ± 0.6**	**4.8 ± 0.7**	**2.9 ± 0.7**
Suns	5.5 ± 0.3	**9.3 ± 0.7**	**7.5 ± 0.4**	**2.5 ± 0.2**
Brown Bears	1.2 ± 0.2	**0.97 ± 0.09**	**0.96 ± 0.15**	**0.82 ± 0.18**
Cheetahs	4.5 ± 0.3	**3.7 ± 0.3**	**3.8 ± 0.3**	**3.7 ± 0.5**

Table 2. Average precision for correlogram descriptors on 15 image classes, using greedily chosen exemplars in a nearest-neighbor classifier. From left to right, columns show the autocorrelogram, color band correlogram with two bands, color band correlogram with three bands. Numbers that differ significantly from the autocorrelogram are bold, and improvements are underlined. Units are percentages; i.e., perfect retrieval = 100.

Class	Auto.	CB2	CB3	GC
Race Cars	6.5 ± 6.2	**0.79 ± 0.25**	**0.73 ± 0.17**	**0.36 ± 0.03**
Wolves	6.5 ± 1.4	6.1 ± 1.9	**5.2 ± 1.9**	**3.0 ± 1.4**
Churches	1.5 ± 0.3	1.1 ± 0.8	1.4 ± 1.1	1.5 ± 1.4
Tigers	26 ± 7	**17 ± 6**	**20 ± 6**	**15 ± 6**
Caves	1.3 ± 0.2	**1.1 ± 0.2**	**1.0 ± 0.1**	**0.59 ± 0.05**
Doors	1.5 ± 0.7	**2.2 ± 1.1**	**2.7 ± 1.5**	**0.95 ± 1.00**
Stained Glass	9.5 ± 7.6	10.0 ± 5.0	**15 ± 5**	**0.32 ± 0.09**
Candy	1.5 ± 0.8	**0.72 ± 0.11**	1.1 ± 0.8	**0.58 ± 0.19**
MVs	2.4 ± 1.1	2.6 ± 1.0	2.5 ± 1.0	**1.4 ± 1.0**
Bridges	1.7 ± 0.8	**1.1 ± 0.2**	**1.1 ± 0.2**	**1.1 ± 0.2**
Swimmers	5.6 ± 5.2	**8.7 ± 4.6**	**8.7 ± 4.7**	**1.3 ± 1.4**
Divers	21 ± 5	**11 ± 4**	**11 ± 4**	**5.2 ± 3.1**
Suns	4.8 ± 1.5	**7.4 ± 2.5**	6.1 ± 2.6	**1.7 ± 0.5**
Brown Bears	2.1 ± 1.7	**0.94 ± 0.47**	**1.2 ± 1.3**	**1.7 ± 1.6**
Cheetahs	6.9 ± 4.4	7.6 ± 3.3	7.6 ± 4.6	**2.8 ± 2.5**

($p < .05$), all the comparisons that achieve significance favor the more complex correlogram versions. This suggests that the boosting process can effectively select the features relevant to the query class, and that giving it more features to work with can enhance this action.

As a practical matter, the fact that $CB3$ can achieve performance near the levels of the general correlogram is encouraging, since it requires only 2% of the storage space. Building a retrieval system based on the general correlogram would be daunting due to its large storage and memory requirements. Thus the future may belong to representations like $CB3$ that combine expressiveness with relative compactness.

Table 3. Average precision for correlogram descriptors on 15 image classes, using boosted classifiers. From left to right, columns show the autocorrelogram, color band correlogram with two bands, color band correlogram with three bands. Numbers that differ significantly from the autocorrelogram are bold, and improvements are underlined. Units are percentages; i.e., perfect retrieval = 100.

Class	Auto.	CB2	CB3	GC
Race Cars	9.4 ± 8.6	19 ± 11	**22 ± 12**	**20 ± 14**
Wolves	2.3 ± 4.2	2.4 ± 3.5	2.6 ± 2.5	1.6 ± 1.4
Churches	0.66 ± 1.25	0.76 ± 1.74	0.57 ± 0.67	0.48 ± 0.84
Tigers	16 ± 10	16 ± 5	15 ± 8	18 ± 8
Caves	0.91 ± 1.18	0.82 ± 1.18	1.7 ± 2.4	5.5 ± 15.9
Doors	2.0 ± 2.9	3.0 ± 4.3	1.5 ± 2.1	1.3 ± 2.1
Stained Glass	44 ± 14	50 ± 12	**55 ± 16**	**64 ± 9**
Candy	12 ± 7	11 ± 9	12 ± 9	8.3 ± 10.5
MVs	1.1 ± 2.0	0.28 ± 0.41	0.23 ± 0.22	0.56 ± 0.52
Bridges	0.084 ± 0.098	1.2 ± 2.7	0.16 ± 0.14	1.9 ± 5.2
Swimmers	13 ± 10	21 ± 5	20 ± 10	**22 ± 8**
Divers	49 ± 13	54 ± 12	50 ± 12	52 ± 12
Suns	29 ± 11	32 ± 8	31 ± 6	30 ± 8
Brown Bears	1.3 ± 3.6	5.7 ± 15.9	0.82 ± 1.23	0.89 ± 1.92
Cheetahs	4.4 ± 4.0	7.2 ± 6.9	11 ± 10	9.7 ± 7.1

4 Conclusion

This paper has systematically examined several variants of the correlogram under a variety of experimental conditions. Boosted classification gives the best average precision over all the experimental frameworks. This result is not unexpected; previous work has shown that boosting improves the retrieval performance of the correlogram [7]. Other work has also shown that boosting can act as a feature selector, choosing features that are correlated with the target class and weeding out those that are not (which might otherwise mislead a classifier by drowning out the significant features) [14]. This paper combines these two

insights by augmenting the standard autocorrelogram with additional features based upon correlations with bands of similar colors. While the new features may not be as relevant for image classification and retrieval as those in the standard autocorrelogram, they can still improve retrieval performance when applied with boosting. This observation, and its experimental confirmation, shows that more remains to be discovered about the humble correlogram.

References

1. V. Athitsos, J. Alon, S. Sclaroff, and G. Kollios. Boostmap: A method for efficient approximate similarity rankings. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, page (to appear), 2004.
2. I.J. Cox, M.L. Miller, T.P. Minka, T.V. Papathornas, and P.N. Yianilos. The bayesian image retrieval system, pichunter: Theory, implementation, and psychophysical experiments. *IEEE Tran. On Image Processing*, 9(1):20–37, 2000.
3. T. G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1924, 1998. Revised December 30, 1997.
4. Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning*, pages 148–156, 1996.
5. J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. Technical report, Dept. of Statistics, Stanford University, 1998.
6. N. Howe. Percentile blobs for image similarity. In *Proceedings of the IEEE Workshop on Content-Based Access of Image and Video Libraries*, pages 78–83, Santa Barbara, CA, June 1998. IEEE Computer Society.
7. N. Howe. A closer look at boosted image retrieval. In *Image and Video Retrieval, Second International Conference*, pages 61–70. Springer, 2003.
8. N. R. Howe. *Analysis and Representations for Automatic Comparison, Classification and Retrieval of Digital Images*. PhD thesis, Cornell University, May 2001.
9. J. Huang. *Color-Spatial Image Indexing and Applications*. PhD thesis, Cornell University, August 1998.
10. J. Huang, S. K. Kumar, M. Mitra, W. Zhu, and R. Zabih. Image indexing using color correlograms. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 1997.
11. F. Jing, M. Li, H. Zhang, and B. Zhang. Support vector machines for region-based image retrieval. In *Proc. IEEE International Conference on Multimedia & Expo, 2003*, 2003.
12. R. E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.
13. R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26(5):1651–1686, 1998.
14. K. Tieu and P. Viola. Boosting image retrieval. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume I, pages 228–235, 2000.