Approximate Image Color Correlograms

Claudio Taranto
Department of Computer Science
University of Bari "Aldo Moro"
via E. Orabona, 4 - Bari, Italy
claudio.taranto@di.uniba.it

Nicola Di Mauro
Department of Computer Science
University of Bari "Aldo Moro"
via E. Orabona, 4 - Bari, Italy
ndm@di.uniba.it

Stefano Ferilli
Department of Computer Science
University of Bari "Aldo Moro"
via E. Orabona, 4 - Bari, Italy
ferilli@di.uniba.it

Floriana Esposito
Department of Computer Science
University of Bari "Aldo Moro"
via E. Orabona, 4 - Bari, Italy
esposito@di.uniba.it

ABSTRACT

The recent explosion in Internet usage and the growing amount of digital images caused by the more and more ubiquitous presence of digital cameras has created a demand for effective and flexible techniques for automatic image retrieval. As the volume of the data increases, memory and processing requirements need to correspondingly increase at the same rapid pace, and this is often prohibitively expensive. Image collections on this scale make performing even the most common and simple image processing and machine learning tasks non trivial. In this paper we present a method to reduce the computational complexity of a widely known method for image indexing and retrieval based on a second order statistical measure. The aim of the paper is twofold: Q1) is it possible to efficiently extract an approximate distribution of the image features with a resulting low error? Q2) how the resulting approximate distribution affects the similarity-based accuracy? In particular, we propose a sampling method to approximate the distribution of correlograms, adopting a Monte Carlo approach to compute the distribution on a subset of pixels uniformly sampled from the original image. A further variant is to sample the neighborhood of each pixel too. Validation on the Caltech 101 dataset proved that the proposed approximate distribution, obtained with a considerable decrease of the computational time, has an error very low when compared to the exact distribution. Result obtained in the second experiment on a similarity-based ranking task are encouraging.

Categories and Subject Descriptors

I.4 [**Computing Methodologies**]: Image Processing And Computer Vision—*Feature Measurement*

General Terms

Experimental, Performance

Keywords

Color correlograms, Feature Extraction and Representation

1. INTRODUCTION

The rapid expansion of digital image libraries has encouraged the development of systems for content-based image retrieval based on indexing and querying engines [7]. In order to index an image the common approach is to extract low level information, such as pixel color, intensity, texture and shape, which could be used to make a feature vector representing the image. This paper addresses the problem of image feature vector computation. To date, the feature vector is calculated analysing each pixel belonging to the image, and thus involving both a huge computational time and a long indexing time. Our approach is to use a method to efficiently compute the image feature vector. We apply a Monte Carlo method to approximate the indexing, analysing a subset of the pixels belonging to an image. We prove that the obtained error is not significantly relevant when compared to the case of considering all the pixels.

The first approach to calculate feature vectors is based on the color histogram [4, 6, 5]. This method is strongly based on pixel colors and it describes for each color level the number of corresponding pixels. For this reason, the image is usually converted in a color space (such as RGB or HSV). The major limit of histogram-based methods is that they only capture global information. It is possible that two semantically different images may correspond to a very similar histogram. An improved statistics is the joint histogram [3] that includes not only color information, but also some other features, such as edge, texture, brightness. This method is more accurate than histogram but it presents the same problem of characterising an image with global information only. In [2], a new approach, named *color correlogram*, combining both global and local image information has been presented. This new statistics describes how pixels with a given color are spatially distributed in the image. The

MM'10, October 25–29, 2010, Firenze, Italy.

correlogram method is generally more accurate and effective than histogram-based methods [2].

The paper is organised as follows. We first briefly review the basic concepts on correlograms and then we present our method to approximate this statistics. Then we conclude the paper with some experimental results and future works.

2. THE PROPOSED APPROACH

A *correlogram* is a statistics expressing how the colors of an image are spatially correlated. A correlogram is defined as a table indexed by color pairs, where the k-th entry for the component (i, j) specifies the probability of finding a pixel of color j at a distance k from a pixel of color i in an image, where k is a distance chosen from the set D.

Let $\mathcal{I}$ be an $n_1 \times n_2$ image, whose colors are quantised into m color classes $\mathcal{C} = \{c_1, ..., c_m\}$ (bins). Give a pixel $p_{xy} \in \mathcal{I}$, $I_{(p)}$ denotes its color and $I_c \triangleq \{p_{ij} \in \mathcal{I} | I_{(p)} = c\}$.

In the following, we assume to use the L_∞-norm to measure the distance beetween pixels. In particular, given $p_{x_1y_1}$ and $p_{x_2y_2}$ two pixel, the distance is computed as follows [2]:

$$|p_{x_1y_1} - p_{x_2y_2}| \triangleq max\{|x_1 - x_2|, |y_1 - y_2|\}. \qquad (1)$$

The *histogram* h of an image $\mathcal{I}$ is defined as $h_{c_i}(\mathcal{I}) = n_1 n_2 \Pr_{p\in\mathcal{I}}[p \in I_c]$, for each color class $c_i \in \mathcal{C}$.

Let $d \in D$ be a distance, the correlogram of an image $\mathcal{I}$ is defined as follows:

$$\gamma^{(d)}_{c_i,c_j}(\mathcal{I}) = \Pr_{p_1\in I_{c_i}, p_2\in I}[p_2 \in I_{c_j} || p_1 - p_2| = d], \qquad (2)$$

where c_i and c_j are two color classes from $\mathcal{C}$, and $d \in D$. The size of the color correlogram matrix is dn_1n_2. A specialisation is the *autocorrelogram* defined as $\alpha^{(d)}_c(\mathcal{I}) = \gamma^{(d)}_{c,c}(\mathcal{I})$.

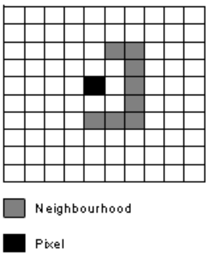

Figure 1: Exact algorithm

In order to calculate the correlogram matrix, given a distance $d \in D$, for each pixel one has to analyse all the pixels belonging to its complete neighbourhood $\mathcal{N}^+_d$, with $|\mathcal{N}^+_d| = 8d$. Here, we propose a method to compute the correlogram by considering only half neighbourhood. Given p_{ij} a pixel whose coordinates are x_i and x_j, and d a distance, we define the neighbourhood $\mathcal{N}^-_d(p_{ij})$ of the pixel p_{ij} as

$$\begin{aligned} \mathcal{N}^-_d(p_{ij}) &= \{(x_{i+k}, y_{j-d})\}_{k=1,\dots,d} \\ &\cup\{(x_{i+d}, y_{j-d+k})\}_{k=1,\dots,2d} \\ &\cup\{(x_{i+d-k}, y_{j+d})\}_{k=1,\dots,d} \qquad (3) \end{aligned}$$

Figure 1 shows with grey pixels the neighbourhood $\mathcal{N}^-_2$ of the black pixel. Hence, proceeding from left to right and from top to bottom, the correlogram $\gamma^{(d)}_{c_i,c_j}(\mathcal{I})$ can be calculated considering $\mathcal{N}^-_d$ also.

2.1 Reducing the computing time

Since the correlogram based approach has been proved to be more effective than histogram, join histogram and autocorrelogram, we have decided to investigate how to decrease the time complexity required to calculate the correlogram. For this reason, we have applied a Monte Carlo approach implemented into two different ways.

2.1.1 The first method: MC1

1. For each distance d, k pixels are randomly selected from the image, with uniform distribution; and then
2. for each selected pixel the neighbourhood is computed and the correlogram matrix accordingly updated.

2.1.2 The second method: MC2

1. For each distance d, k pixels are randomly selected from the image, with uniform distribution; and then
2. for each selected pixel p, h pixels are randomly selected from the complete neighbourhood $\mathcal{N}^+_d(p)$.

3. VALIDATION

In order to validate the proposed approach, the Calltech101 dataset [1] has been used. It regards images, whose size is roughly 300x200 pixels, belonging to 101 different categories, with a mean of 50 images for each category. We performed two experiments. The former aimed to verify whether the proposed approximate methods are able to compute correlograms with a low error with respect to the exact distribution. While, the latter experiments show preliminary results regarding the retrieval accuracy rate comparing both the exact and approximate correlogram distributions. Images have been mapped to color spaces of 16 and 64 bins.

3.1 Indexing based error

In this experiment, for each class one random image has been selected. For each of these images all the algorithms MC1, MC2 and Exact (that compute the exact correlogram matrices) have been applied in order to compute the correlogram matrices along the distances 1, 3, 5 and 7.

In order to obtain the error made by the approximate methods, a baseline value is computed considering the error obtained by a random matrix containing for each cell a random number among the interval [0,1], whose rows and columns sum to 1. Given C_E the exact correlogram matrix, C_{MC} the approximate correlogram matrix computed with a method among MC1 and MC2, and C_R the random matrix, the *gain* is defined as

$$G = 1 - e_{MC}/e_R, \qquad (4)$$

where

$$e_X = \sqrt{\sum_{ij}(C_E(i, j) - C_X(i, j))^2/|C_E|}.$$

Figure 3 shows experimental results, these are obtained by averaging across all the 101 images in the dataset. The graphs report the results using 16 and 64 colors bins, and let to vary percentage of the sampled pixels from 10% to 50% for MC1 and MC2, and sampling the 50% of the neighbourhood for MC2. The gain adopting 16 bins is better than that

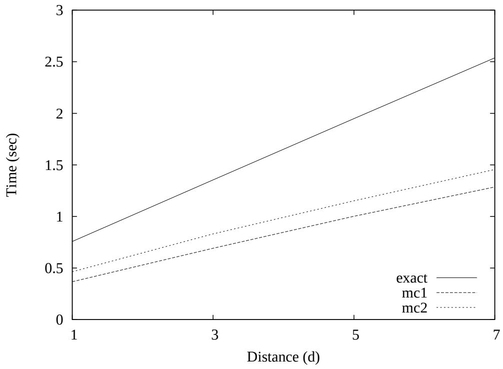

Figure 2: Time (MC1 50% pixel, and MC2 50% neighbourhood).

obtained with 64 bins, probably due to the fact that with few bins it is more simple to approximate the exact distribution. Furthermore, Figure 3 reports the error e_{MC1} using 16 colors bins with compared to the error e_R both with the distance equal to 7. Figure 2 reports the time, averaged over all the considered images, to compute a correlogram matrix with the three algorithms. As we can see, we can affirmatively response to the question Q1, since both approximate methods turn out to be faster than the exact method. MC2 takes slightly longer because it calculates the list of the random neighbours to be analysed, while MC1 just considers all the neighbourhood.

3.2 Similarity based error

In this second experiment, five classes, with 200 images from each one, have been selected, and specifically airplanes, motorbikes, faces, watch and leopard. In order to measure the error made during the retrieval phase, for each image the most similar one has been identified. Each image in our system is represented by the correlograms calculated on 4 distances (1,3,5 and 7). The similarity between two images, $\mathcal{I}$ and $\mathcal{I}'$, is calculated as follows, as reported in [2]:

$$|\mathcal{I}-\mathcal{I}'| := \sum_{c_i,c_j\in\mathcal{C},d\in D} \frac{|\gamma^{(d)}_{c_i,c_j}(\mathcal{I}) - \gamma^{(d)}_{c_i,c_j}(\mathcal{I}')|}{1+\gamma^{(d)}_{c_i,c_j}(\mathcal{I})+\gamma^{(d)}_{c_i,c_j}(\mathcal{I}')}. \qquad (5)$$

This process has been done by comparing each image with all the others and ranking the results adopting Equation 5, both with the exact and the approximate method. In order to evaluate the similarity-based error obtained with MC1 and MC2, the results obtained by the exact method are considered as the ground truth. Tables 1, 2, 3 and 4 report the results of this experiment, where %Pixel indicates the percentage of pixels examined for each image, %Neighbourhood represents the percentage of pixels in the considered neighbourhood and Depth is the number of images considered as a reference in comparison. In particular, let M be a method, $\mathcal{S}$ a set of images, and $\{\mathcal{Q}_1,\ldots,\mathcal{Q}_n\}$ a set of query images. For a query $\mathcal{Q}$, let $\mathcal{A}_M$ be the list of images similar to $\mathcal{Q}$ sorted according to the metric reported in Equation 5 and obtained adopting the method M. Let us to define $\mathrm{rank}_M(\mathcal{I})$, with $\mathcal{I}\in\mathcal{A}_M$, the relative position of $\mathcal{I}$ in the ordered list $\mathcal{A}_M$.

Hence, given n queries, and denoting with E the exact method and with MC a Monte Carlo approach:

- Depth1 is the percentage of images that both the exact

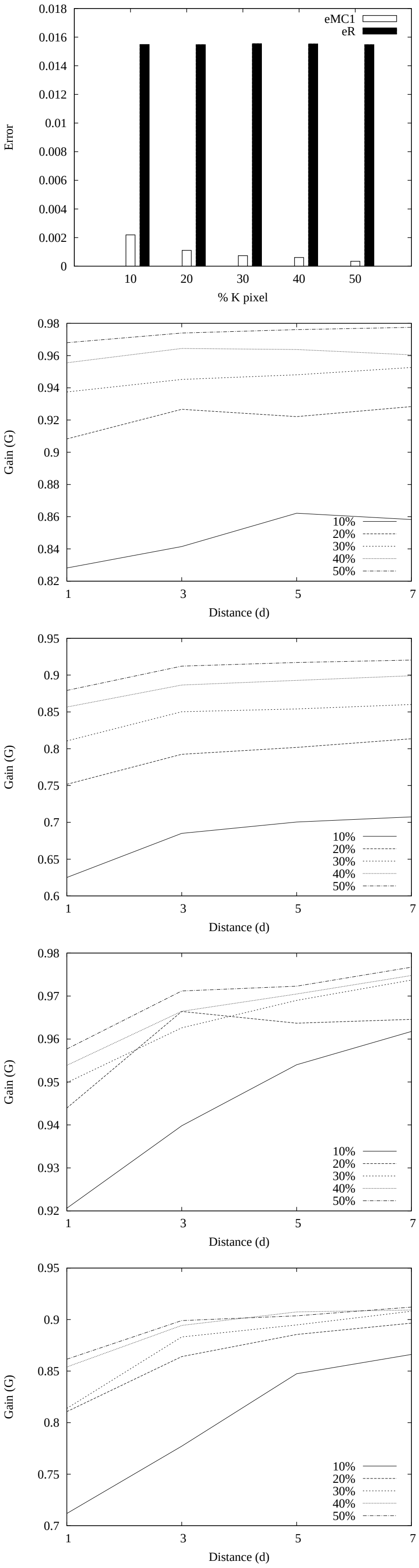

Figure 3: MC1 and random error with d=7 (top); MC1 gain (from the top: second with 16 bins, and third with 64 bins); MC2 gain (from the bottom: second with 16 bins and 50% neighbourhood, and first with 64 bins and 50% neighbourhood).

		Depth%		
%Pixel	%Neighbourhood	1	2	3
2	100	34.2	49.7	56.9
5	100	48.9	65.2	73.9
10	100	55.3	71.5	80.4
20	100	63.2	78.1	85.4
30	100	64.7	79.3	85.7
40	100	68.4	84.2	90.3
50	100	74.5	86.8	91.5

Table 1: MC1 16 bins similarity based error.

		Depth%		
%Pixel	%Neighbourhood	1	2	3
2	10	29.2	43.6	52.8
2	30	35.9	46.6	54.5
2	50	31.4	46.8	53.5
5	10	48.6	64.9	73.5
5	30	45.4	63.4	71.3
5	50	48.2	64.9	73.2
10	10	53.1	69.7	77.1
10	30	54.8	70.8	79.2
10	50	54.2	70.9	79.6
20	10	58.2	74.9	82.7
20	30	62.8	77.6	85.7
20	50	61.1	77.2	85.3
30	10	61.5	78.0	85.2
30	30	61.9	78.4	86.8
30	50	66.3	80.5	87.8
40	10	64.9	79.1	86.2
40	30	67.5	81.6	88.3
40	50	67.8	81.3	87.5
50	10	67.2	81.9	88.2
50	30	68.2	82.1	88.8
50	50	71.7	85.3	90.5

Table 2: MC2 16 bins similarity based error.

		Depth%		
%Pixel	%Neighbourhood	1	2	3
10	100	28.8	44.0	52.5
30	100	38.6	58.8	67.9
50	100	39.5	59.9	69.5

Table 3: MC1 64 bins similarity based error.

		Depth%		
%Pixel	%Neighbourhood	1	2	3
10	10	29.8	45.1	52.7
10	30	27.1	43.7	52.6
10	50	24.8	42.0	51.7
30	10	38.6	58.8	67.9
30	30	38.6	57.0	67.0
30	50	38.8	58.1	68.4
50	10	40.9	60.5	69.9
50	30	43.8	62.8	72.3
50	50	43.7	62.5	71.9

Table 4: MC2 64 bins similarity based error.

and the approximate algorithm rank as the first best answer; i.e. $\sum_i \text{rank}_E(\mathcal{Q}_i) = \sum_i \text{rank}_{MC}(\mathcal{Q}_i) = n$;

- Depth2 is the percentage of images that the approximate algorithm ranks as the first best answer and the exact algorithm ranks in the firstly two best positions; i.e. $\sum_i \text{rank}_{MC}(\mathcal{Q}_i) = n$ and $\text{rank}_E(\mathcal{Q}_i) \leq 2$;
- Depth3 is the percentage of images that the approximate algorithm ranks as the first best answer and the exact algorithm ranks in the firstly three best positions; i.e. $\sum_i \text{rank}_{MC}(\mathcal{Q}_i) = n$ and $\text{rank}_E(\mathcal{Q}_i) \leq 3$.

As we can see from the tables, satisfactory results have been obtained using 16 bins and the MC1 algorithm. As previously reported, with 64 bins the algorithm is not able to correctly approximate the exact distribution, and, hence the results are satisfactory. Finally, this slightly confirms the question Q2.

4. CONCLUSION AND FUTURE WORK

In this work we presented a method to reduce the computational complexity to compute color correlograms adopting a Monte Carlo approach to compute the distribution on a subset of pixels uniformly sampled from the original image. We showed that it is possible to efficiently extract an approximate distribution of the image features with a resulting low error, and that the resulting approximate distribution slightly affects the similarity-based accuracy. As a future work, we will investigate the applicability of other approximate methods and the possibility to use compression methods to obtain similar results.

5. REFERENCES

[1] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. In *Workshop on Generative-Model Based Vision*, 2004.

[2] Jing Huang, S. Ravi Kumar, Mandar Mitra, Wei-Jing Zhu, and Ramin Zabih. Image indexing using color correlograms. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 762–768. IEEE, 1997.

[3] Greg Pass and Ramin Zabih. Comparing images using joint histograms. *Multimedia Systems*, 7(3):234–240, 1999.

[4] Raimondo Schettini, Gianluigi Ciocca, and Isabella Gagliardi. Content-based color image retrieval with relevance feedback. In *Proceedings of the IEEE International Conference on Image Processing*, pages 75–79, 1999.

[5] Michael J. Swain and Dana H. Ballard. Color indexing. *International Journal of Computer Vision*, 7:11–32, 1991.

[6] Leonid Taycher, Marco La Cascia, and Stan Sclaroff. Image digestion and relevance feedback in the imagerover www search engine. In *Proceedings of International Conference on Visual Information*, pages 85–94, 1997.

[7] Adam Williams and Peter Yoon. Content-based image retrieval using joint correlograms. *Multimedia tools and Applications*, 34:239–248, 2007.