On the computation of correlation based image features

Simon *Vogl*
Telecooperation Dept., University of Linz, Austria
Technical Report 2000-0901

September 25, 2000

Abstract

Color correlograms have been presented as an efficient color feature for image databases. The attempt to use this feature for video indexing revealed a more efficient way to compute color correlograms than the original solution. A family of related features is presented.

1 Introduction

Color correlograms have been presented in [2] as a promising image feature for the registration of pictures in an image database and content based retrieval.

The presented algorithm has a high running time complexity that greatly reduces its usability, and has therefore not been used in video indexing or other time-critical applications.

In this paper, we present a different solution that overcomes these problems, and makes a whole field of *different* image features accessible.

The rest of this article is organized as follows: First we recapitulate how color correlograms are defined and computed, then we present our approach and a short comparison. Finally, we show some interesting new features that can be computed easily with our algorithm.

2 Color Correlograms

This image feature computes the propability for pixels of a specific color c_i that neighbouring pixels (in a distance d) have color c_j. For the following formal definition, the set I_c is introduced as a convinient short form that contains all pixels in I that are colored with c ($I_C := \{p|I(p) = c\}$):

$$\gamma^{(d)}_{c_i,c_j}(I) = \Pr_{p_1 \in I_{c_i}, p2 \in I}\Big(p_2 \in I_{c_j}, |p_1 - p_2| = d\Big) \tag{1}$$

The authors of [2] define the *autocorrelogram* as the correlation of one color with itself:

$$\alpha^{(d)}_c(I) = \gamma^{(d)}_{c,c}(I) \tag{2}$$

As a computationally efficient distance measure, the L_∞-norm is used:

$$|p_1 - p_2| := max(|x_1 - x_2|, |y_1 - y_2|)$$

As a trade-off between accuracy and storage size, the authors propose to compute auto-correlograms for all colors and for a set of distances $d \in D$. This reduces storage needs from $O(m^2|D|)$ to $O(m|D|)$ (where m is the number of colors).

For the actual computation, the authors introduce intermediate quantities:

$$\lambda^{c,h}_{(x,y)}(d) = |\{(x+i,y) \in I_c | 0 \leq i \leq d\}| \tag{3}$$

$$\lambda^{c,v}_{(x,y)}(d) = |\{(x,y+i) \in I_c | 0 \leq i \leq d\}|$$

These are used to compute a quantity that is closely related to the correlogram, and is defined by

$$\Gamma^{(d)}_{c_i,c_j}(I) = \Big|\{p_1 \in I_{c_i}, p2 \in I_{c_j} \mid |p_1 - p_2| = d\}\Big| \tag{4}$$

where $\gamma^{(d)}_{c_i,c_j}(I) = \Gamma^{(d)}_{c_i,c_j}(I)/(h_{c_i}(d) * 8d)$. The actual value is computed by adding the proper values of the intermediate matrices:

$$\begin{aligned}\Gamma^{(d)}_{c_i,c_j)} &= \sum_{(x,y)\in I_{c_i}} \Big(\lambda^{c_j,h}_{(x-k,y+k)}(2d) + \lambda^{c_j,h}_{(x-k,y-k)}(2d) \\ &+ \lambda^{c_j,v}_{(x-k,y-k+1)}(2d-2) + \lambda^{c_j,v}_{(x+k,y-k+1)}(2d-2)\Big)\end{aligned}$$

Due to the nature of the intermediate quantities running time of the overall algorithm is $O(n^2 d)$, where n^2 is the number of pixels.

3 An alternative approach

The attempt to use color correlograms as a feature for video segmentation and indexing showed that the algorithm presented by the original authors needs a good amount of processing time. The authors point out that the main problem is their intermediate construct they use: Computation time grows linear with the distance d the correlogram shall be computed for. They propose two different approaches to save computing time, depending on the magnitude of d, but stay roughly in the same order of complexity.

3.1 Cumulative matrices

We introduce a different data structure to overcome this issue: Our algorithm makes use of a *cumulative matrix* as a helper construct that is invariant to the changing distance.

Each cell $p(x,y)$ of a cumulative matrix I^{cum} contains the sum of all cells of the original matrix I that lie in the upper left rectangle of the cell:

$$I^{cum}(x,y) = \sum_{i \leq x,\, j \leq y} I(i,j) \tag{5}$$

The matrix may be built using an incremental approach:

$$I^{cum}_c(x,y) = I^{cum}_c(x-1,y) + I^{cum}_c(x,y-1) - I^{cum}_c(x-1,y-1) + I_c(x,y)$$

For cells outside the matrix itself, the following rules apply. This can be verified by placing the original matrix in a plane of zeroes.

$$\begin{aligned}I^{cum}_c(x,y) &= 0, \text{ if } x < 0 \text{ or } y < 0 \\ I^{cum}_c(x,y) &= I^{cum}_c(x,h) \text{ if } y > h \\ I^{cum}_c(x,y) &= I_c(w,y) \text{ if } x > w\end{aligned}$$

Using the binary images I_c as the original matrix, a cumulative 'image' I_c^{cum} can be computed and used to count the number of pixels in an arbitrary rectangular area defined by the points $p_1(x_1, y_1)$ and $p_2(x_2, y_2)$:

$$\rho_{I_c}(p_1, p_2) = I_c^{cum}(x_1, y_1) - I_c^{cum}(x_1, y_2 - 1) - I_c^{cum}(x_2 - 1, y_1) + I_c^{cum}(x_2 - 1, y_2 - 1) \quad (6)$$

As we are interested in quadratic areas, weuse an alternative definition of equation 6 that uses a single point and a radius d:

$$\begin{aligned} \rho_{I_c}(x, y, d) \quad = \quad & I_c^{cum}(x + d, y + d) - I_c^{cum}(x - d - 1, y + d) \\ & - I_c^{cum}(x + d, y - d - 1) + I_c^{cum}(x - d - 1, y - d - 1) \end{aligned} \quad (7)$$

3.2 Computing correlograms

In our video indexing system, we compute a cumulative matrix for each color $I_{c_i}^{cum}$and derive the number of pixels for a fixed distance by using equation 7 to compute the total count:

$$\Gamma_{ci,cj}^{(d)}(I) = \sum_{p \in I_{c_i}} \rho_{I_{c_j}}(x_p, y_p, d) - \rho_{I_{c_j}}(x_p, y_p, d - 1) \quad (8)$$

Note that the last entry of the cumulative matrix (the one to the lower right) contains the number of pixels that have the specified color, or, in other words, the value of the histogram for that color. This can be easily used to compute the final correlogram or stored as a second feature for comparison.

4 Evaluation

Looking at the definition of our helper construct (eqn. 5)we use, it is easy to see that the running time for its computation is $O(n^2)$ - or linear to the number of pixels. Due to its invariance to the distance d, the cumulative matrix for a color may be computed once and reused for the whole range of distances $d \in D$.

Furthermore, the complexity of equation 8 is also $O(n^2)$, so the total complexity is also linear to the number of pixels. It can be also seen that the distance sets may contain larger values without impacting performance ($D = \{1, 2, 4, 8, 16, 32\}$ for example), so larger image properties may also be taken into account.

5 Derived features

By varying several details in equation 8, a whole range of different image features can be created. In the following paragraphs, we will present some of these, but note that they still need to be evaluated regarding their performance for video segmentation or image retrieval.

Note that it influences retrieval and segmentation performance if the scaling of the individual entries by $\frac{1}{h_{c_i}}$ is added - we present only the associated pixel counts, but the relation is analog to γ and Γ.

5.1 Correlogram stripes

To overcome noise and motion, it may be favorable to have a more relaxed view on the distance. Using thicker stripes than only the one pixel wide line in the original formula might compenstate for these phenomena.

$$\Phi_{ci,cj}^{(d,f)}(I) = \sum_{p \in I_{c_i}} \rho_{I_{c_j}}(x_p, y_p, d) - \rho_{c_j}(x_p, y_p, d-f) \tag{9}$$

In [1] a new image feature is introduced, called *banded color correlogram*, that can be seen as a special case. The following count is equvalent:

$$B_{ci,cj}^{(d)}(I) = \sum_{p \in I_{c_i}} \rho_{I_{c_j}}(x_p, y_p, d) \tag{10}$$

5.2 Color density measures

As a variant to the autocorrelogram, the following count computes the density of pixels for a given distance:

$$\Delta_c^{(d)}(I) = \sum_{p \in I} \rho_{I_{c_j}}(x_p, y_p, d) - \rho_{I_{c_j}}(x_p, y_p, d-1) \tag{11}$$

Note that this feature is nothing less than the sum of all color correlograms for a specific color: $\Delta_c^{(d)}(I) = \sum_{c_j} \gamma_{c,c_j}^{(d)}(I)$

5.3 Inter-frame correlation

Especially for video-indexing applications, it could be useful to investigate inter-frame color correlations. Care has to be taken, though, as this is inherently motion-sensitive, but may provide interesting results for analysing intra-shot activity.

5.4 Arbitraty Rectangular areas

The use of cumulative matrices opens another aspect: It is possible to calculate color densities (or correlations) not only for quadratic, but also for rectangular areas. This could lead to image features that are sensitive to tilts: Using for example a $11x3$-recangle around a point, horizontal areas should result in peak values - rotation should result in a decrease for the horizontal, but in an increase for the corresponding vertical filter kernel.

6 Finally

We showed a new approach to efficiently compute a promising image feature. This opens the way for further study as the new algorithm can be applied to a whole range of related image features.

References

[1] J. Huang, S. Kumar, and R. Zabih. An automatic hierarchical image classification scheme, 1998.

[2] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih. Image indexing using color correlograms. In *IEEE Computer Vision and Pattern Recognition Conference*, page B7: Video and image database indexing, 1997.