Consistency Check of Image Databases

Szabolcs Sergyán, László Csink

John von Neumann Faculty of Informatics
Budapest Tech
Nagyszombat u. 19, H-1034 Budapest, Hungary
{sergyan.szabolcs, csink.laszlo}@nik.bmf.hu

Abstract: Checking the consistency of image databases is not a completely solved problem. To decide if an image has already been inserted into the database can only be checked by actually looking through the images, or using the textual descriptive keywords attached to the database items. Both the visual checking and the keyword search in a large image database may result in errors.

Several methods of content-based image retrieval [3,5,6,9] and image clustering [8] are known that could be used for determining image database consistency. However; most of these have drawbacks and are sensitive for errors. Thus it seems reasonable to develop a robust, relatively fast algorithm that can identify very similar images of a large database, where very similar means that the two images are probably the same, maybe taken in different illumination conditions. The authors present their Matlab solution that has been tested on a database of 250 color flags used in [4].

Keywords: Image Database Consistency, Color Object Recognition

1 Introduction

Interest in image databases has grown considerably in the last years. Locating a desired image in a large image database is thus a typical task. Problems with textual image indexing resulted in more and more interest in retrieving images of automatically derived features based on color, texture or shape – Content Based Image Retrieval.

Before effective search in an image databese is performed, it is best to guarantee that the database is consistent in the sense that the same image is included only once, or more precisely, each image contains only one version, or at least the different versions of an image are detected. Consistency of versions is an important issue regarding object-oriented databases [10]. In CAD applications, a database often stores different alternatives of the same object; these databases are called multiversions, otherwise the database is monoversion.

Visual consistency is considered in [11] in a different sense in the framework of multisource visual information processing, with the goal to reduce complexity and to resolve ill-posed problems.

In our contribution we develop a robust method for checking color image database consistency. The experimental database[1] contains n=272 flags, thus the consistency check involves n*(n-1)/2 comparisions, so a robust algorithm was selected.

2 Project Description

The project is based on the following algorithm. Take an RGB image of the database and convert it to grayscale format [2]. Then, using k-means clustering [1], transform it into a grayscale image having $k = 4$ gray values (Figure 1). In the following, identify the connnected regions in the image where the grayvalue is identical. It may be thought that the thus found regions are the regions of the same color in the original image. This is, however, not alwys true, as red and green will be transformed into the same gray value. Color normalization [7] may help in this case, as then normalized red and normalized green will be different.

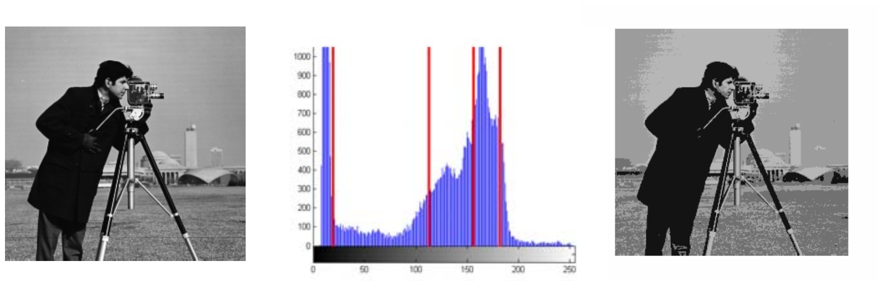

Figure 1
(a) is the well-known cameraman; (b) is the histogram of (a), the thick lines are the centers of the clusters; (c) is a grayscale image with the 4 grayscale value

Now consider another approach. Transform the RGB image into a socalled indexed image with a palette of 10-20 colors. A connected set of pixels will be called homogeneous if more than 98% of the pixels have the same indexed color, and the region will be called to be of this color.

---

[1] http://www.flags.net

Let us define a similarity metric between two images *f* and *g* as follows. Let $r_f(1)$, $r_f(2)$, $r_f(2)$,…, $r_f(p)$ denote homogeneous regions of *f*, where $r_f(1)$.area > $r_f(2)$.area > $r_f(3)$.area >….> $r_f(p)$.area, and *p* is a fixed constant. Denote $r_f(i)$.color the color of the *i*th region. Then *f* and *g* are similar if and only if

$$g \approx f \text{ iff } d_1 = \sum_{j=1}^{p} \left| r_f(j).area - r_g(j).area \right| < \varepsilon_1 \text{ and}$$

$$d_2 = \sum_{j=1}^{p} \left| r_f(j).color - r_g(j).color \right| < \varepsilon_2 \text{, where } p = \min\left(6, N_f, N_g\right) \text{ and}$$

N_f is the number of homogen regions in *f*.

There are, however, flags that have several regions of roughly the same size (e.g. Hungarian or Italian flags, that have even the same colors as well, the difference being in the setting of the colors). If we compare two such gflags, then the small variety in the sizes of the color stripes enhances greatly the success of comparision. Therefore, the topology of the color regions also need to be considered.

3 Experiments

The test were run on a database containing 272 flags, which were digitally drawn. The flags typically contained no t too many rekatively large homogeneous regions, as well as some smaller objects.

In the first phase we were looking for the optimal choice of ε_1 and ε_2 (see above). To this aim, for each flag in the database, we found the nearest (non-identical) flag using the above distance. Then we defined ε_1 and ε_2 as the minima of the resepctive columns of Table 1.

	$\min(d_1)$	$\min(d_2)$
Afghanistan	0.0004363	0.40308
Albania	0.02696	0.0039216
Alderney	0.12244	0
⋮		
Yemen	0	0
Zambia	0.041875	0.037163
Zimbabwe	0.018716	0

Table 1
Minima of distances

Having determined the parameters, the consistency check of the database was possible. Table 2 lists the items in the database to which a similar other item was found.

No.	Tested flag	Similar item in database	comment
1.	Burma	Myanmar	same flag
2.	Chad	Romania	same flag
3.	CocosIslands	Heard&McDonaldIslands	same flag
4.	Commonwealth	Kazahstan	they look similar
5.	Congo-Kinshasa	EuropeanUnion	yellow stars on blue, but with different structure
6.	France	FrenchGuiana	same flag
7.	France	Guadeloupe	same flag
8.	France	Martinique	same flag
9.	France	Mayotte	same flag
10.	France	Tromelin	same flag
11.	FrenchGuiana	Guadeloupe	same flag
12.	FrenchGuiana	Martinique	same flag
13.	FrenchGuiana	Mayotte	same flag
14.	FrenchGuiana	Tromelin	same flag
15.	Greenland	Indonesia	they look simililar
16.	Greenland	Monaco	same flag
17.	Greenland	Poland	they look similar
18.	Guadeloupe	Martinique	same flag
19.	Guadeloupe	Mayotte	same flag
20.	Guadeloupe	Tromelin	same flag
21.	Indonesia	Monaco	same flag
22.	Indonesia	Poland	same flag
23.	*IsleofMan*	*Somalia*	*totally different*
24.	Martinique	Mayotte	same flag
25.	Martinique	Tromelin	same flag
26.	Mayotte	Tromelin	same flag
27.	Monaco	Poland	same flag
28.	NewZealand	Tokelau	same flag
29.	NordicCouncil	SouthKorea	they look similar
30.	*SaudiArabia*	*WesternEuropeanUnion*	*totally different*

Table 2

Result of the consistency check

4 Results and Conclusions

According to the test, out of $\binom{272}{2}$ comparisons only in two cases were the results totally wrong (case 23. and 30., Table 2.) Similarities were found in cases 4, 5, 15, 17 and 29, which are justified if we look at the flags of the correponding countries.

Identity was found in several cases. In case 1. the reason is that Burma and Myanmar are two fifefrent names of the same country, so this item appears in the database twice. Case 2. is just a coincidence, Chad and Romania simply have the same flag. In cases 6-14, the reason of the same flag is clearly historical.

We think that our robust method is useful for the check of consistency in some image databases, where relatively big, homogeneous color regions are typical. We do not say that our method is error free, but the errors were minimal in the test. Clearly other aspects than color regions could be considered, but then the running time will considerably incerase.

References

[1] Bradley, P. S., and Fayyad, U. M.: Refining Initial Points for K-Means Clustering. In: Proc. of 15th International Conference on Machine Learning, Morgan Kaufmann, San Fracisco, 1998, pp. 91-99

[2] Buchsbaum, P.: A Spatial Processor Model for Object Colour Perception. J. Franklin Inst., Vol. 310, 1980, pp. 1-26

[3] Chen, Y., Wang, J. Z., and Krovetz, R.: CLUE: Cluster-based Retrieval of Images by Unsupervised Learning, IEEE Transactions of Image Processing, 2004

[4] Csink, L., and Sergyán, Sz.: Recognition of Color Flags. Universität Koblenz-Landau, Scientific talk, 2004

[5] Kiss, A., Németh, T., Sergyán, Sz., Vámossy, Z., and Csink, L.: Recognition of a Moving Object in a Stereo Environment Using a Content Based Image Database. In: Proc. of 3rd Slovakian-Hungarian Joint Symposium on Applied Machine Intelligence, Herl'any, Slovakia, January 21-22, 2005, pp. 65-74

[6] Matas, J.: Colour-based Object Recognition. PhD dissertation, University of Surrey, 1996

[7] Paulus, D., Csink, L., and Niemann, H.: Color Cluster Rotation. In: Proc. of International Conference on Image Processing, 1998, pp. 161-165

[8] Pavan, M., and Pelillo, M.: A New Graph-Theoretic Approach to Clustering and Segmentation. CVPR, 2003

[9] Sergyán, Sz.: Special Distances of Image Color Histograms. In: Proc. of 5th Joint Conference on Mathematics and Computer Science, Debrecen, Hungary, June 9-12, 2004, pp. 92

[10] Cellary, W., and Jomier, G.: Consistency of Versions in Object-Oriented Databases. In Proc. 16th VLDB, pages 432--441, 1990

[11] Pinz, A.: Consistent Visual Information Processing Applied to Object Recognition, Landmark Definition, and Real-Time Tracking. VMV'01, Stuttgart, Germany, 2001. http://citeseer.ist.psu.edu/pinz01consistent.html