World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:2, No:11, 2008

A New Method in Detection of Ceramic Tiles Color Defects using Genetic C-Means Algorithm

Mahkameh S. Mostafavi

Abstract**—**In this paper an algorithm is used to detect the color defects of ceramic tiles. First the image of a normal tile is clustered using GCMA; Genetic C-means Clustering Algorithm; those results in best cluster centers. C-means is a common clustering algorithm which optimizes an objective function, based on a measure between data points and the cluster centers in the data space. Here the objective function describes the mean square error. After finding the best centers, each pixel of the image is assigned to the cluster with closest cluster center. Then, the maximum errors of clusters are computed. For each cluster, max error is the maximum distance between its center and all the pixels which belong to it.

After computing errors all the pixels of defected tile image are clustered based on the centers obtained from normal tile image in previous stage. Pixels which their distance from their cluster center is more than the maximum error of that cluster are considered as defected pixels.

Keywords**—**C-Means algorithm, color spaces, Genetic Algorithm, image clustering.

I. Introduction

THE classical c-means clustering algorithm (CMA) is a well-known clustering technique in the field of pattern recognition. It is an iterative scheme which starts from an initial distribution of cluster centers in data space. Each data point is assigned to the cluster with closest cluster center, after which each cluster center is updated as the center of mass of all data points belonging to that particular cluster. This procedure is repeated until convergence [4].

As is well known but neglected by many researchers is that CMA is very much dependent on the choice of the initial distribution of cluster centers. In this way, the algorithm ends up in a local optimum, which in real life applications can be far away from the real global optimum, especially when a large number of data points and clusters is involved.

A way to deal with local optimality is to use stochastic optimization schemes. Invented in the early 70's, Genetic Algorithm (GA) is become more and more popular over the last years. A GA is inspired by biological evolution, and widely believed to be an effective global optimization algorithm.

An implementation of a genetic algorithm begins with a population of (typically random) chromosomes. One then evaluates these structures and allocates reproductive opportunities in such a way that those chromosomes which represent a better solution to the target problem are given more chances to "reproduce" than those chromosomes which are poorer solutions. The "goodness" of a solution is typically defined with respect to the current population [7].

Genetic operations such as crossover and mutation are defined. The mutation operator changes individual elements of a string, the crossover operator interchanges parts between strings. The combination of these operations is then repeated during several generations. The intrinsic parallelism of a genetic algorithm, i.e. the ability of manipulating in parallel large numbers of strings, and the crossover operation whereby good portions of different strings are combined both make the technique a very effective optimization technique.

II. The Genetic C-Means Algorithm (GCMA)

The classical CMA is a clustering algorithm which optimizes an objective function based on a measure between the data points and the cluster centers in the data space. When M data points in a N-dimensional space are clustered into C clusters, the objective function can be chosen as:

$$\sigma^2 = \frac{1}{M}\sum_{i=1}^{C}\sum_{x\in C_i}(x - v_i)^2 \qquad (1)$$

Where ***x*** is a N-dimensional vector containing the coordinates of data point x and ***vi*** the N dimensional vector containing the coordinates of the center of cluster i, Ci is the collection of data points belonging to cluster i. This objective function describes the mean squared error (MSE) when replacing each data point by the center of the cluster to which it belongs. Minimizing the objective function with respect to ***vi*** leads to the following conditions:

$$V_i = \frac{\sum_{x\in C_i} x}{\sum_{x\in C_i} 1} \qquad (2)$$

i.e. the cluster centers are positioned at the center of mass of the data points belonging to the cluster. The second set of

Manuscript received November 14, 2006.

Mahkameh S. Mostafavi is with Department of Electrical Engineering, Yazd University, Yazd, P.O.Box 89195-741, Iran (phone: +98 351 8211670-9 Ext 2387; fax : +98 351 82110699; e-mail : mk_mostafavi@yahoo.com).

conditions that minimizes the MSE states that a data point should be associated with the closest cluster center.

$$x \in C_j \;\; j = Min_i[(x - v_i)^2] \tag{3}$$

A set of cluster centers which satisfies (2) and (3), satisfies minimal objective function conditions and is called a local optimum. One way to satisfy both sets of equations is by using CMA. Here, one starts with an initial set of cluster centers $\{\vec{v}_1,.....\vec{v}_C\}$. The cluster centers can be chosen randomly, or as the output of another procedure, e.g. a splitting procedure. Each data point is assigned to the cluster with closest cluster center using (3). The position of the cluster centers is then updated using (2). This procedure is repeated until convergence. This iterative scheme is known to converge sufficiently fast. However more than one solution satisfies (2) and (3). Depending on the initial choice of cluster centers, the algorithm ends up in a local optimum that can be far away from the global one. In this paper the problem of local optima is solved by developing a hybrid algorithm combining CMA and a genetic approach. When using genetic algorithms, a population of genetic strings (binary or other) is generated. A fitness function is defined which assigns to each string a value of fitness. Through genetic evolution, strings regenerate, mutate and interchange information. Since these genetic operations favorite strings with high fitness value, the population evolves to an optimal one. A genetic design for clustering can be constructed in the following way:

- The number of clusters is fixed to a value C.
- A population of P random strings is generated. Each string r represents a set of C cluster centers $\{\vec{v}_1,.....\vec{v}_C\}_r$. Notice that these strings are strings of vectors, the elements of which are the coordinates of a cluster center in data space. In practice, each coordinate is put into a separate string, so that each vector string is in fact a structure of several component strings:

$$\{\vec{v}_1,.....,\vec{v}_C\} = (\{v_1^1,.....,v_c^1\},\{v_1^2,.....,v_C^2\},.....,\{v_1^N,....v_C^N\}) \tag{4}$$

Henceforth we will refer to the vector string as string. All operations on a string are in fact performed on each component string separately. Notice also that the strings are not binary strings, as is the case in many genetic algorithm applications. Our strings contain integer or floating point values. Three genetic operators are defined:

Regeneration: On each string of the population MSE is calculated using (1). The inverse of the MSE is used as fitness function. All strings are pair wise compared. The string with lowest MSE is copied into the other.

Crossover: On each string $\{\vec{v}_1,.....\vec{v}_C\}$ one-point crossover is applied with probability Pc (Pc=0.8) (i.e. a uniform random number r between 0 and 1 is generated and crossover is applied if r < Pc). Out of the total population a partner string $\{\vec{v}_1^{'},.....\vec{v}_C^{'}\}$ is randomly chosen (i.e. a uniform integer random number i between 1 and P is generated, and the string i is chosen as a partner string). Then an integer random number j between 1 and C is generated. Both strings are cut in two portions at position j and the portions $\{v_{j+1},.....,v_C\}$ are mutually interchanged, i.e.:

$$\begin{aligned} \{\vec{v}_1,.....,\vec{v}_C\} &\rightarrow \{\vec{v}_1,....,\vec{v}_j,\vec{v}^{'}{}_{j+1},...\vec{v}_C^{'}\} \\ \{\vec{v}_1^{'},.....,\vec{v}_C^{'}\} &\rightarrow \{\vec{v}_1^{'},....,\vec{v}_j^{'},\vec{v}_{j+1},...,\vec{v}_C\} \end{aligned} \tag{5}$$

Mutation: Mutation is performed on each element $\vec{v}_j$ of each string with a small probability Pm (Pm=0.05). From $\vec{v}_j$ one of the N component is chosen at random. Then a random number, taking the binary values -1 or 1, is generated, and is added to the chosen component.

The total of these three operations is called a generation. After a generation, the string with lowest MSE is stored. Then the next generation is performed. If after the next generation a string with lower MSE is created, it replaces the stored one. The string which is stored after G generations is chosen as the optimal result. After applying several generations, the stored string approaches an optimum. For problems with high dimensionality, large populations may have to be defined, and a large number of generations may be necessary before the system converges.

To reduce the search space drastically, the genetic approach is combined with CMA. Hereby CMA is applied to all strings of the population during each generation, before the regeneration step. By doing this, all strings are forced into local optima. This procedure will be referred to as the genetic c-means algorithm (GCMA).

In this paper GCMA is used as an algorithm to find the best cluster centers which will be used in clustering the image of a normal tile. This algorithm is applied as follows:

--First some initial sets of colors are generated. These sets are randomly chosen over the color space. Each color consists of its red, green, and blue component and has the meaning of a cluster center in color space. Here the CIEL*U*V* color space is used. The reason of using this space is that the distance between any two points in RGB does not give the measure of color difference between two color perceptions. Also CIE is a system in which colors can be clustered due to Human Visual System (HVS).

-- Each string of centers chosen in first stage is fed into CMA separately after which the genetic operators are applied on the resulting color strings. This is repeated during G generations and results in best cluster centers [4].

III. Applying the Proposed Algorithm to the Tile Images

After finding the best cluster centers using GCMA, the pixels of a normal tile image are assigned to the cluster with closest cluster center. Then the maximum error of clusters is computed. For each cluster the maximum error is the maximum distance between its center and all the pixels which belong to it. This is done using equation (3) or the equation below:

$$error = \sqrt{(v_x - c_x)^2 + (v_y - c_y)^2 + (v_z - c_z)^2} \quad (6)$$

Where (v_x, v_y, v_z) are the color coordinates of each point belongs to a cluster. And (c_x, c_y, c_z) are the color coordinates for the center of that cluster. After computing errors, all the pixels of defected tile image are clustered based on the centers obtained from the image of normal tile in previous stage. A pixel which its distance from cluster center is more than the maximum error of that cluster is considered as a defected pixel.

During the use of CMA in different stages of GCMA algorithm, a faster version of CMA, named randomized CMA can be applied. To this end a small subset of data points are is chosen randomly, after which one iteration step of CMA is applied on this subset, i.e. each data point of the subset is assigned to its closest cluster center after which the cluster centers are updated as the centers of mass of the data points of the subset which are assigned to the cluster. This procedure is repeated for several times using different subsets. Typical values are 10 subsets of a few hundred data points. The result of this randomized approximation of CMA leads to suboptimal results which are sufficient for the population to produce useful information to interchange and to allow for the production of a more optimal solution.

IV. Experiments and Discussion

Automatic visual inspection of color textures plays a crucial role in machine vision applications. In this paper an approach is shown for the detection of surface defects in color textured images using genetic and c-means clustering algorithms. Since this approach extremely depends on resolution of pictures, taking images of tiles should be done with high resolution CCD cameras or very sensitive scanners. In addition the light sources should be established in appropriate places to make a symmetric luminescence on different directions of tile surface [9].

The proposed algorithm can be used for any different type of tiles such as single colored, simple (without any pattern), patterned…

References

[1] C. Boukovalas, F. De Natale, G. De Toni, J. Kittler,, R. Marik, M. Mirmehdi, M. Petrou, P. Leroy, R. Salgari and G. Vernazza, "ASSIST: Automatic System for Surface Inspection and Sorting of Tiles", Journal of Materials Processing Technology, 82(1-3): 179—188, oct 1998.

[2] MACS TECH, ASPECT: Automatic Selector Processing for Ceramic Tiles, Jun 1999.

[3] Adrian Ford, Alan Roberts," Colour Space Conversions", August 11, 1998, http:// www.poyton.com/PDFs/coloreq.pdf

[4] P. Scheunders, A Genetic C-means Clustering Algorithm Applied to Color Image Quantization", Pattern recognition, 859-866, 1997.

[5] Rafael C. Gonzales, Richard E. Woods, Digital Image Processing, Second Edition, Prentice-Hall, Inc. 2002.

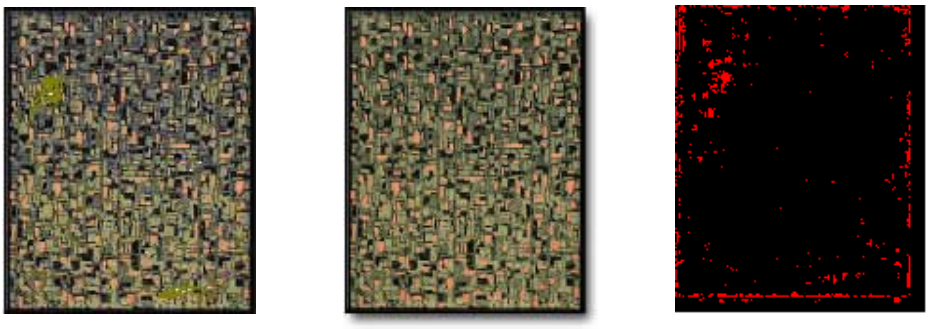
[6] Robin Biesbroek, GA Tutorial homepage, http://www.estec.esa.nl/outreach/gatutor/contents.htm

[7] Darrell Whitely, GA Tutorial, http://www.samizdat.mines.edu/ga_tutorial/

[8] J. Kittler, R. Marik, M. Mirmehdi, M. Petrou, J. Song, "Detection of Defects in Color Textured Surfaces", In: IAPR Proc. of Machine Vision Applications 94, pages 558—567, December 1994.

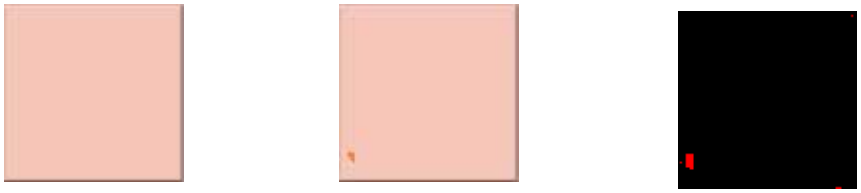
[9] C. Boukouvalas, J. Kittler, R Marik, B. and M. Petrou., "Automatic Grading of Ceramic Tiles Using Machine Vision "IEEE in Symposium on Industrial Electronics, 13-18 1994.

[10] B. Thomas, M.Mirmehdi and Xianghua Xie. "Inspecting color tonality on Textured Surfaces" Proceedings of 1st International Conference on Image Analysis and Recognition, pages 810-817.Springer LNCS 3212, September 2004.



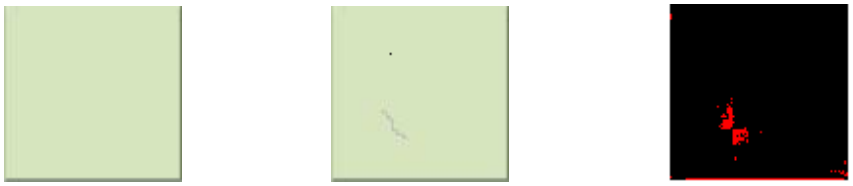
(a) (b) (c)

Fig. 1 (a) Perfect granite tile (b) Defected tile (c) Detected defects



(a) (b) (c)

Fig. 2 (a) Perfect single color tile (b) Defected Tile (c) Detected defects



(a) (b) (c)

Fig. 3 (a) Perfect single color tile (b) Defected Tile (c) Detected defects

(a) (b) (c)

Fig. 4 (a) Perfect patterned tile (b) Defected Tile (c) Detected Defects