FACE COMPARISION USING PSO ALGORITHM

Dr.A.V.Senthilkumar[#1], J.Savitha[*2]

[#]Director, Master of Computer Application, Hindustan Arts and Science College, Coimbatore, Tamilnadu, India

[*]Phd Scholar, Karpagam University, Coimbatore,Tamilnadu,India

Abstract

It is used specially for the compression of images where tolerable degradation is required. With the wide use of computers and consequently need for large scale storage and transmission of data, efficient ways of storing of data have become necessary. With the growth of technology and entrance into the Digital Age, the world has found itself amid a vast amount of information. Dealing with such enormous information can often present difficulties. Image compression is minimizing the size in bytes of a graphics file without degrading the quality of the image to an unacceptable level. The reduction in file size allows more images to be stored in a given amount of disk or memory space. It also reduces the time required for images to be sent over the Internet or downloaded from Web pages.JPEG and JPEG 2000 are two important techniques used for image compression.

JPEG image compression standard use DCT (DISCRETE COSINE TRANSFORM). The discrete cosine transform is a fast transform. It is a widely used and robust method for image compression. It has excellent compaction for highly correlated data.DCT has fixed basis images DCT gives good compromise between information packing ability and computational complexity.

JPEG 2000 image compression standard makes use of DWT (DISCRETE WAVELET TRANSFORM). DWT can be used to reduce the image size without losing much of the resolutions computed and values less than a pre-specified threshold are discarded. Thus it reduces the amount of memory required to represent given image.

Keywords— Put your keywords here, keywords are separated by comma.

Introduction

Face Recognition is an effective pathway between human and computer, which has a lot of applications in information security, human identification, security validation, law enforcement, smart cards, access control and etc. For this reasons, industrial and academic computer vision and pattern recognition researchers have a significant attention to this task.

As the PSO equations given above work on real numbers, a commonly used method to solve discrete problems is to map the discrete search space to a continuous domain, to apply a classical PSO, and then to demap the result. Such a mapping can be very simple (for example by just using rounded values) or more sophisticated.[39]

However, it can be noted that the equations of movement make use of operators that perform four actions:

- computing the difference of two positions. The result is a velocity (more precisely a displacement)
- multiplying a velocity by a numerical coefficient
- adding two velocities
- applying a velocity to a position
- Face components such as eyes, nose, mouth or facial templates such as nose length and width, mouth position, and chin type. These features are used to recognize an unknown face by matching it to the nearest neighbor in the stored database. Statistical features extraction is usually driven by algebraic methods such as principal component analysis (PCA), and independent component analysis (ICA) [6]. These methods find a mapping between the original feature spaces to a lower dimensional feature space. The shortage of PCA is that it treats inner-class and out-classes equally [3], [4], [5] and therefore it is sensitive to light and changes of expressions. LDA has higher performance than PCA in face recognition but the traditional LDA cannot provide reliable and robust solution since their separable criterion is not relevant to classification precision. Alternative algebraic methods are based on transforms such as down sampling, Fourier transforms (FT), discrete cosine transforms (DCT), and the discrete wavelet transforms (DWT). Transformation based feature extraction methods
such as the DCT and DWT were found to generate good FR accuracies with very low computational cost [8]. DCT is one of the approaches used in image compressing which is also used to extract features [9], [10]. Wavelet analysis has both a good qualities in time domain and frequency domain which is an ideal tool in analyzing unsteady signals.

Algorithm

Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a swarm intelligence technique developed by Dr. Eberhart and Dr. Kennedy in 1995 [1]. In PSO, the swarm consists of particles which move around the solution space of the problem. These particles search for the optimal solution of the problem in the predefined solution space till the convergence is achieved.

A basic variant of the PSO algorithm works by having a population (called a swarm) of candidate solutions (called particles). These particles are moved around in the search-space according to a few simple formulae. The movements of the particles are guided by their own best known position in the search-space as well as the entire swarm's best known position. When improved positions are being discovered these will then come to guide the movements of the swarm. The process is repeated and by doing so it is hoped, but not guaranteed, that a satisfactory solution will eventually be discovered.

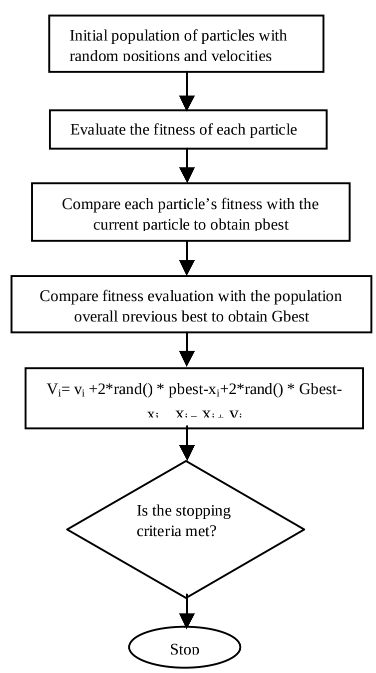
Formally, let $f: \mathbb{R}^n \to \mathbb{R}$ be the fitness or cost function which must be minimized. The function takes a candidate solution as argument in the form of a vector of real numbers and produces a real number as output which indicates the fitness of the given candidate solution. The gradient of f is not known. The goal is to find a solution $\mathbf{a}$ for which $f(\mathbf{a}) \le f(\mathbf{b})$ for all $\mathbf{b}$ in the search-space, which would mean $\mathbf{a}$ is the global minimum. Maximization can be performed by considering the function $h = -f$ instead.

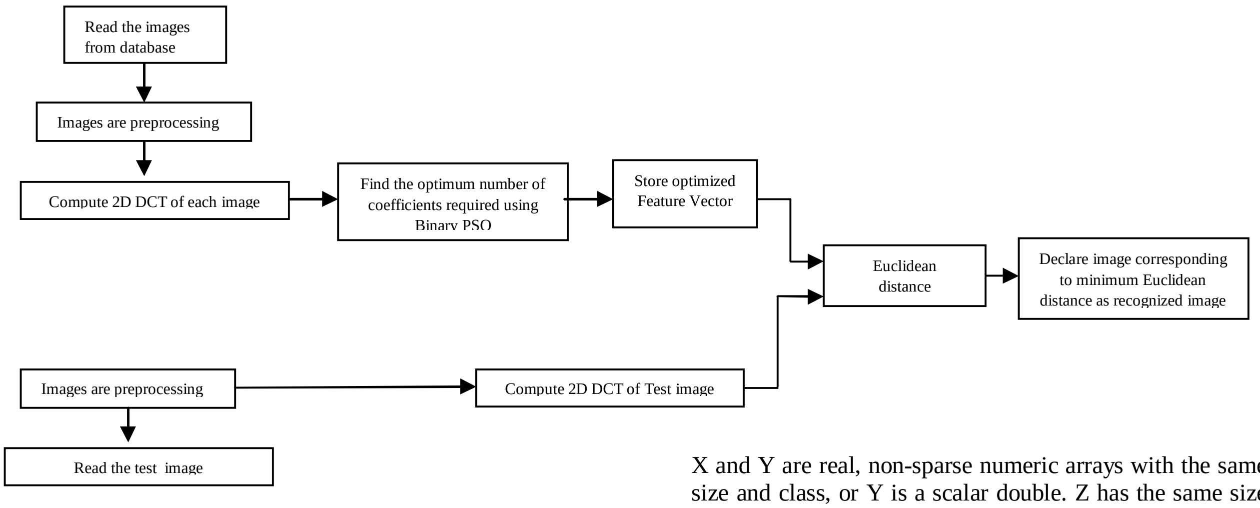
Let S be the number of particles in the swarm, each having a position $\mathbf{x}_i \in \mathbb{R}^n$ in the search-space and a velocity $\mathbf{v}_i \in \mathbb{R}^n$. Let $\mathbf{p}_i$ be the best known position of particle i and let $\mathbf{g}$ be the best known position of the entire swarm. A basic PSO algorithm is then:

- For each particle $i = 1, ..., S$ do:
  - Initialize the particle's position with a uniformly distributed random vector: $\mathbf{x}_i \sim U(\mathbf{b}_{lo}, \mathbf{b}_{up})$, where $\mathbf{b}_{lo}$ and $\mathbf{b}_{up}$ are the lower and upper boundaries of the search-space.
  - Initialize the particle's best known position to its initial position: $\mathbf{p}_i \leftarrow \mathbf{x}_i$
  - If $(f(\mathbf{p}_i) < f(\mathbf{g}))$ update the swarm's best known position: $\mathbf{g} \leftarrow \mathbf{p}_i$
  - Initialize the particle's velocity: $\mathbf{v}_i \sim U(-|\mathbf{b}_{up}-\mathbf{b}_{lo}|, |\mathbf{b}_{up}-\mathbf{b}_{lo}|)$
- Until a termination criterion is met (e.g. number of iterations performed, or adequate fitness reached), repeat:
  - For each particle $i = 1, ..., S$ do:
    - For each dimension $d = 1, ..., n$ do:
      - Pick random numbers: $r_p, r_g \sim U(0,1)$
      - Update the particle's velocity: $\mathbf{v}_{i,d} \leftarrow \omega\, \mathbf{v}_{i,d} + \varphi_p\, r_p\, (\mathbf{p}_{i,d}-\mathbf{x}_{i,d}) + \varphi_g\, r_g\, (\mathbf{g}_d-\mathbf{x}_{i,d})$
    - Update the particle's position: $\mathbf{x}_i \leftarrow \mathbf{x}_i + \mathbf{v}_i$
    - If $(f(\mathbf{x}_i) < f(\mathbf{p}_i))$ do:
      - Update the particle's best known position: $\mathbf{p}_i \leftarrow \mathbf{x}_i$
      - If $(f(\mathbf{p}_i) < f(\mathbf{g}))$ update the swarm's best known position: $\mathbf{g} \leftarrow \mathbf{p}_i$
- Now $\mathbf{g}$ holds the best found solution.

The parameters ω, φ_p, and φ_g are selected by the practitioner and control the behaviour and efficacy of the PSO method.

Face Comparison





ORL Face Database

Database	PPR1	PPR2	PPR3	PPR4	No of Training	No. of Test
ORL	impyramid	imadjust	Edge	imadd	4	6

Referring to above table, we define the terms as follows:

Impyramid (I, direction): Impyramid (I, direction) computes a Gaussian pyramid reduction or expansion of I by one level, direction can be 'reduce' or 'expand'.

2)Imadjust(I,[],[],gamma): J = imadjust(I,[lowin; highin];[lowout; highout],gamma) maps the values in I to new values in J, where gamma specifies the shape of the curve describing the relationship between the values in and J. If gamma is less than 1, the mapping is weighted toward higher (brighter) output values. If gamma is greater than 1, the mapping is weighted toward lower (darker) output values. If you omit the argument, gamma defaults to 1 (linear mapping).

3)Edge(I,canny): J = edge(I) takes a gray scale or a binary image I as its input, and returns binary image J of the same size as I, with 1's where the function finds edges in I and 0's elsewhere. edge (I,'canny') specifies the Canny method.

4)Uint8 (I): uint8(I) returns the stored integer value of object as a built-in uint8. If the stored integer word length is too big for a uint8, or if the stored integer is signed, the returned value saturates to uint8.

5)Imadd(I1,I2): Z = imadd(X,Y) adds each element in array X with the corresponding element in array Y and returns the sum in the corresponding element of the output array Z. X and Y are real, non-sparse numeric arrays with the same size and class, or Y is a scalar double. Z has the same size and class as X, unless X is logical, in which case Z is double.

6) Fspecial (‘sobel’): h = fspecial('sobel') returns a 3-by-3 filter h that emphasizes horizontal edges using the smoothing effect by approximating a vertical gradient. If it’s needed to emphasize vertical edges, transpose the filter ‘h’.

7)Imfilter (I, H,’same’): Filters the multidimensional array I with the multidimensional filter H. The array I can be logical or a non-sparse numeric array of any class and dimension. The result has the same size and class as I.'Same' means the output array is the same size as the input array. This is the default behavior when no output size options are specified.

There are 10 different images of 40 distinct subjects. For some of the subjects, the images were taken at different times, varying lighting slightly, facial expressions (open/closed eyes, smiling/non-smiling) and facial details (glasses/no-glasses). All the images are taken against a dark homogeneous background and the subjects are in up-right, frontal position (with tolerance for some side movement).

The files are in PGM format and can be conveniently viewed using the 'xv' program. The size of each image is 92x112, 8-bit grey levels. The images are organised in 40 directories (one for each subject) named as:

sX

where X indicates the subject number (between 1 and 40). In each directory there are 10 different images of the selected subject named as:

Y.pgm

where Y indicates which image for the specific subject (between 1 and 10).

Experiments and Results

Database	DCT	20*20	
ORL		unmodified	Modified
	Recognization	92.5	93.5
	Non Zero Coeffieient	218	50

Conclusion

Preprocessing of images and fine tuning of binary PSO parameters were found to yield better results in both ORL databases in terms of recognition rate and number of DCT coefficients in the final subset (Non zero coefficients).Number of DCT coefficients was very less in ORL as images were resized to half. Thus proper use of image preprocessing depending upon image sets helps in increasing the recognition rate as well as reducing the number of DCT coefficients in the final subset. Future studies can be made to implement the face recognition system in real time

References

[1] J. Kennedy and R. Eberhart, Particle Swarm Optimization. Proceedings of the IEEE International Conference on Neural Networks, vol. 6, pp. 1942–1948, Piscataway, NJ, December 1995.

[2] J. Kennedy and R. C. Eberhart. A Discrete Binary Version of the Particle Swarm Algorithm.Proc. IEEEInternational Conference on Systems, Man, and Cybernetics, vol. 5, pp. 4104-4108, Oct. 1997.

[3] W. Zhao, R. Chellappa, P. J. Phillips and A. Rosenfeld. Face recognition: A literature survey. ACM Computing Surveys, 35(4):399–458, 2003.

[4] Gregory Shakhnarovich and Baback Moghaddam. Face recognition in subspaces. Technical Report TR2004-41, Mitsubishi Electric Research Laboratories, 201 Broadway, Cambridge, Massachusetts 02139, 2004.

[5] A. S. Tolba, A.H. El-Baz, and A.A. El-Harby. Face Recognition: A Literature Review. International Journal ofSignal Processing, vol. 2, no. 2, pp. 88-103. 2006

[6] Rabab M. Ramadan and Rehab F. Abdel Kader. Face Recognition using Particle Swarm Optimization-based selected features. International Journal of Signal Processing, Image Processing and Pattern Recognition, 2(2):51-66, 2009.

[7] Ziad M. Hafed and Martin D. Levine. Face Recognition using the Discrete Cosine Transform. International Journal of ComputerVision, 43(3):167–188, 2001.

[8] F. M. Matos, L. V. Batista, and J. Poel. Face Recognition Using DCT Coefficients Selection. Proc. of the 2008 ACM Symposium on Applied Computing, (SAC'08),pp. 1753-1757, March 2008.

[9] Dr H B Kekre, Tanuja Sarode, Prachi Natu and Shachi Natu. Performance comparison of Face Recognition using DCT against Face Recognition using Vector Quantization Algorithms LBG, KPE, KMGC, KFCG. International Journal Of Image Processing, 4(3):377–389,2010.