Computer Graphics Identification Using Genetic Algorithm

Wen Chen[1], Yun Q. Shi[1], Guorong Xuan[2], Wei Su[1]
[1]*Department of Electrical and Computer Engineering*
New Jersey Institute of Technology, Newark, NJ, USA
[2]*Department of Computer Science, Tongji University, Shanghai, China*
wc47@njit.edu, shi@njit.edu

Abstract

This paper proposes the use of genetic algorithm to select an optimal feature set for distinguishing computer graphics from digital photographic images. Our previously developed approach has derived a 234-D feature vector from each test image in HSV color space. The statistical moments of characteristic functions of the image and its wavelet subbands were selected as the distinguishing image features. Since it is possible that only certain image features contain significant information with respect to the classification, the image features with insignificant contributions to classification may be eliminated to reduce the dimensionality of the feature vectors while maximizing the classification performance. Famous for its efficiency in searching the optimal solution in a very large space, the genetic algorithm is applied to find a reduced feature set which consists of only 100-D features per image in our investigation. The experimental results have demonstrated that the 100-D reduced feature set outperforms the 234-D full feature set.

1. Introduction

Computer graphics are created by a variety of rendering software. Due to the advancement in rendering technique, computer graphics come to appear so photorealistic that it may be used as a convincing form of photographic image forgery. As in Figure 1 where three computer graphics are illustrated, it is very challenging for people to point out with fully confidence that these images are computer graphics instead of photographic images. Here we define the photographic images as the direct output of imaging acquisition devices such as digital camera.

Figure 1. Examples of computer graphics

Several prior research works have been done to separate computer graphics from photographic images [1, 2, and 3]. In our prior work [3], the statistical moments of characteristic function of the image, its prediction error and their wavelet subbands are selected as distinguishing features which are extracted in HSV color space. The dimensionality of feature vector is 234. The experimental results indicated that the classification performance was not greatly degraded if we only used 156-D features extracted in H and V components, meaning that the redundant or unnecessary features may exist. Since the 156-D feature set is only one of all feature combinations of 2^{234}, it is highly expected that an optimal feature set exists among all the feature combinations.

Theoretically, we can apply the brute force approach to each possible feature set in search of the optimal one. However, it is impractical to perform an exhaustive search for the desired feature set in the space of all feature combinations if the feature dimension is very high. In the scenario where we have 234 features, the total number of combinations would be as large as 2^{234}. Since the genetic algorithm (GA) can quickly find the optimal solution in a large space, we propose the use of GA [4] to find the optimal feature set to maximize the classification performance.

The rest of this paper is organized as follows. Section 2 reviews the distinguishing features proposed in [3]. The feature selection using binary GA is presented in Section 3. In Section 4, the experimental results are demonstrated. Finally, the conclusions are drawn in Section 5.

2. Image features

To accurately distinguish computer graphics from photographic images, formulation of image features is a critical step. We have presented the feature definition and their extraction procedure in the prior work [3]. We summarize them here.

The characteristic function (CF) in the context is defined as the Fourier transform of the image (or its wavelet subbands) histogram. The statistical moments of the CFs of both a test image and its wavelet subbands are selected as features. Denote the CF by $H(f_j)$, the statistical moment is defined as follows.

$$M_n = \sum_{j=1}^{(N/2)} f_j^n \left| H(f_j) \right| \bigg/ \sum_{j=1}^{(N/2)} \left| H(f_j) \right| \qquad (1)$$

where $H(f_i)$ is the CF component at frequency f_i, n is the moment order, and N is the total number of points in the horizontal axis of the histogram.

In addition, we extract features in the same manner from the prediction-error image and its wavelet subbands. The prediction-error image is the difference between the test image and its prediction which is constructed by the following prediction algorithm [5].

$$\hat{x} = \begin{cases} \max(a,b) & c \le \min(a,b) \\ \min(a,b) & c \ge \max(a,b) \\ a+b-c & otherwise \end{cases} \qquad (2)$$

where a, b, c are the context of the pixel x under considerations. $\hat{x}$ is the prediction value of x. The locations of a, b, c are illustrated as in Figure. 2.

x	b
a	c

Figure 2. Prediction context

To collect image features, i.e. moments of characteristic functions, a test color image of either a computer graphic or a photographic image is represented by the H, S and V components. Each component image is first decomposed into three levels based on, say, Daubechies or Haar wavelet. At each level i, $i = 1, 2, 3$, there are four subbands (LL, LH, HL and HH). Totally, 13 subbands are involved in the feature extraction if the component image itself is considered as a subband at level 0. For each subband, the first three moments are derived according to Equation (1), resulting in 39 features. For the prediction- error image of each component image, the same wavelet decomposition is employed and another 39 features are collected. Eventually 78 features are extracted from each component image and its prediction-error image. Since there are three component images and three prediction-error

component images, the total number of features is 234. The block diagram of feature generation procedure is shown in Figure.3 where the component is H, S and V, respectively. The prediction error of H, S and V are computed according to Equation (2).
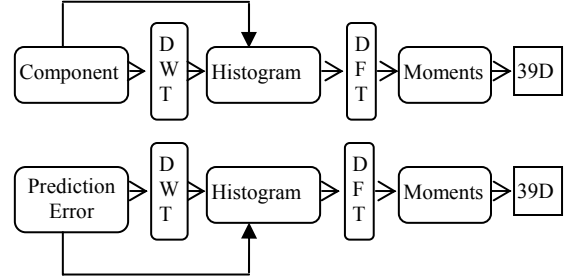

Figure 3. Feature extraction

3. Feature selection via binary GA

3.1. Feature selection strategy

The classification performance heavily relies on the features used. Too few features will provide insufficient information to achieve satisfactory classification accuracy while too many features will increase computational time and degrade classification performance in certain situations. For example, it has been shown that the performance of SVM classifiers may be degraded in situations where irrelevant features are included [6]. This leads to the need for the feature selection strategy to find an optimal feature subset from a given large set of features. Well-known for its ability to offer efficient search of potential solutions in a large space, the GA is a suitable candidate for selecting an optimal feature subset.

With GA, the search is implemented as an iterative procedure which can start with an arbitrary population which consists of a certain number of possible feature combinations. If the search technique is properly designed and implemented, a satisfactory solution will be quickly found in the limited number of iterations. The feature combination with the best fitness in each iteration is saved. After the iterative procedure stops, all of the saved feature combinations are analyzed to determine each feature's contribution to the classification performance. Only the features with the significant contribution to classification performance will be selected to construct the optimal feature subset which is eventually used in the classification system. In summary, the feature selection strategy is composed of search for the fittest feature combinations and analysis of each feature's contribution to classification performance. We will first describe the binary GA in

this section. The determination of contribution significance will be illustrated in Section 4.

3.2. Binary GA

Developed by John Holland in 1975, the GA is an optimization and search algorithm based on the principles of genetics and natural selection [7]. The GA provides potential solutions to the optimization problem in the evolutionary way. The evolution is implemented as an iterative process in computer simulation. In the GA, the representations of potential solutions to an optimization problem are called individuals or chromosomes. If each individual in the population is represented in binary as a string of 1s and 0s, the GA is called binary GA. The search of potential solutions usually starts with an initial population of randomly generated individuals and evolves in generations. In each generation, the fitness of each individual is evaluated by the fitness function. The fittest individuals are selected to reproduce offsprings through genetic operators such as crossover and mutation. The search terminates when a set number of iterations is exceeded or an acceptable solution has been reached. The iterative process is illustrated in Figure 4.
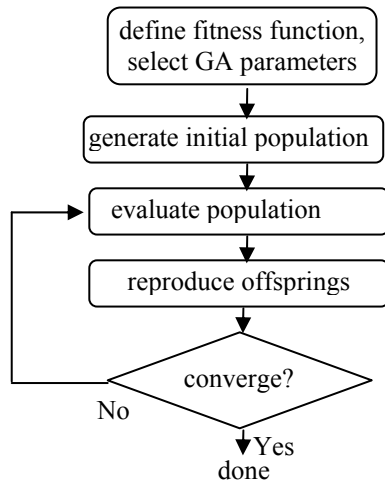
Figure 4: The iterative procedure of GA

The individual's binary representation and fitness function are two main issues in the implementation of a binary GA. In the feature selection problem, one possible feature combination is determined by the individual represented by a binary string:

$$individual = [b_1 b_2 b_3 \cdots b_{n-2} b_{n-1} b_n] \qquad (3)$$

where n is the feature vector dimension, and b_i ($i=1,n$) is a binary bit "1" or "0" to determine the feature selection. The value "1" of b_i denotes inclusion of the

ith feature, and the value "0" denotes rejection of the ith feature. For example, suppose only b_1, b_2 and b_3 take on values of "1", the feature subset will include the first three elements from the n-D full feature vector.

The fitness of each individual in the population is evaluated through a complete training/testing cycle in this investigation. The training data with the selected features are used to train the SVM classifier. The performance of the trained classifier is evaluated using the testing data. All individuals are evaluated in the above process in each generation. The survival is based on the classification performance of each individual. In this study, we will simply use the classification accuracy of testing data as the fitness.

4. Experiments

To facilitate the classification performance comparison between the full feature set and the optimal feature subset, we used the same image data set in [3] which include 1900 photographic images and 800 computer graphics in the Columbia Image Dataset [8].

The following GA parameters are selected: population size = 96, selection rate = 0.5 and mutation rate = 0.25. The single-point crossover is applied. Based on the parameter settings, from the 96 individuals in the population, only half of them (i.e. 96 x 0.5) will survive to the next generation. We chose to mutate 25% of the population except for the fittest individual. In this case the number of mutations is:

$$\#of\ mutations = \mu \times (N-1) \times 234 \qquad (4)$$

where μ and N is mutation rate and population size, respectively. 234 is the number of the features extracted from a given color image. When mutation occurs, a randomly selected mutation point will be changed from a "1" to a "0", and visa versa. The crossover point was selected by a random number. The number of the generations is set to be 300.

The Support Vector Machine (SVM) classifier with poly kernel was employed in evaluating the fitness of each individual [9]. To train the classifier, we randomly selected the training samples to include 5/6 of image set (1580 photographic mages and 665 computer graphics). 135 photographic images and 135 computer graphics not involved in training were used for classification performance (or fitness) evaluation. We changed the seed of random number generator in every five generation to avoid the use of the same training-testing data set in evaluating each possible feature subset.

To speed up the search, the parallel computation was performed on Warewulf clusters using pMatlab

[10] which provides an efficient mechanism for running Matlab programs on parallel computers. The processor speed of each node is 2.4 GHz. We performed the search procedure with 16 cluster nodes each of which was responsible for evaluating six individuals in each generation.

The search procedure was iterated by 300 generations. The fittest individual in each generation was saved. Eventually 300 fittest individuals were kept. To determine the significance of each feature's contribution to classification, 100 best individuals are chosen from 300 saved individuals to constitute a 100x234 matrix as in figure 5 where the element $b_{i,j}$ takes on bit "1" or bit "0".

$$\begin{bmatrix} b_{1,1} & b_{1,2} & \cdots & b_{1,233} & b_{1,234} \\ b_{2,1} & b_{2,2} & \cdots & b_{2,233} & b_{2,234} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ b_{99,1} & b_{99,2} & \cdots & b_{99,233} & b_{99,234} \\ b_{100,1} & b_{100,2} & \cdots & b_{100,233} & b_{100,234} \end{bmatrix}$$

Figure 5: The 100 best individuals

The significance of each feature's contribution to the classification is measured by the probability of the occurrence of "1" in each column. The higher the probability in the ith (i=1, 2, …, 234) column, the more significant the ith feature's contribution to the classification is. A preset threshold T is used to select the features. The threshold is set to be 0.5. The ith feature is selected if the probability is greater than 0.5. Otherwise, the ith feature is excluded in the feature subset. We found 100 columns with probabilities greater than the threshold. Eventually the GA-selected feature subset includes 100 features which consist of 39, 26, and 35 features from H, S and V component, respectively.

The classification performance of the 100 selected features was evaluated in the 20-run training-testing cycles. In each run, 1580 photographic mages and 665 computer graphics were randomly selected to train the SVM classifier, 135 photographic images and 135 computer graphics not involved in the training was used to test the trained classifier. For comparison purpose, the training and testing data set in each run was selected with the same random number seed as in [3]. The classification performance for 100-D feature subset and 234-D full feature set is listed in Table 1 where TPR (true positive rate) represents the detection rate of computer graphics, TNR (true negative rate) represents the detection rate of photographic images, and the ACC is the average detection rate. The results show that the GA-selected feature subset outperforms in average the full feature set. The experiment demonstrated the efficiency of binary GA. The optimal feature subset was found from only about $2^{14.8}$ ($\approx$300x96) instead of 2^{234} feature combinations.

Table 1. SVM classification performance

	TPR	TNR	ACC
100-D	73.4%	91.2%	82.3%
234-D	71.9%	92.3%	82.1%

5. Conclusions

We have proposed the application of the binary GA to feature selection in computer graphics identification. The feature selection is based on the significance of each feature's contribution to the classification performance. With the parallel computation, the binary GA quickly found an optimal 100-D feature set. The experimental results have demonstrated the performance improvement by the reduction of feature dimensionality.

References

[1] S. Lyu and H. Farid, "How realistic is photorealistic?" *IEEE Transactions on Signal Processing*, 53, pp. 845-850, February 2005.
[2] T.-T. Ng, S.-F. Chang, J. Hsu, L. Xie, and M.-P. Tsui, "Physics-motivated features for distinguishing photographic images and computer graphics," in *ACM Multimedia*, Singapore, November 2005.
[3] W, Chen, Y. Q. Shi and G. Xuan, "Identifying computer graphics using HSV color model and statistical moments of characteristic functions," *IEEE International Conference on Multimedia and Expo* (*ICME07*), Beijing, China, July 2-5, 2007.
[4] Goldberg, D.E., *Genetic Algorithms in Search, Optimization and Machine Learning,* Reading Mass., Addison-Wesley, 1989.
[5] M. Weinberger, G. Seroussi, and G. Sapiro, "LOCO-I: A low complexity content-based lossless image compression algorithm," *Proc. of IEEE Data Compression Conf.* pp. 140-149, 1996.
[6] O. Barzilay and V.L. Brailovsky, "On domain knowledge and feature selection using a support vector machine," *Pattern Recognition Letters,Vol.20 ,No 5,*pp.475-484,1999
[7] Holland, J.H., *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbors, 1975.
[8] Columbia University DVMM Research Lab: Columbia Photographic Images and Photorealistic Computer Graphics Dataset.
[9] C. C. Chang and C.J. Lin, LIBSVM: a library for support vector machines, 2001
[10] MIT Lincoln Laboratory, pMatlab: Parallel Matlab Toolbox http://www.ll.mit.edu/pMatlab/