AN APPROACH FOR EYE DETECTION IN IMAGES WITH UNCONSTRAINED BACKGROUNDS IN REAL-TIME USING PARALLEL GENETIC ALGORITHM

A. Çağatay Talay
e-mail: ctalay@itu.edu.tr
Istanbul Technical University, Faculty of Electrical & Electronics Engineering, Department of Compute Engineering, 34469, Maslak, Istanbul, Turkey

Key words: Eye Detection, Parallel Genetic Algorithm

ABSTRACT

In this paper, a new reliable method for detecting human eyes in an arbitrary image used for real-time eye tracking is devised. The approach is based on searching the eyes with Parallel Genetic Algorithm. As the genetic algorithm is a computationally intensive process, the searching space for possible face regions is limited to possible eye regions so that the required timing is greatly reduced. The algorithm works on complex images without constraints on the background, skin color segmentation and so on. The proposed system performs eye detection and tracking in real-time. The eye detection process works predictably, fairly, reliably and regardless of the perspective. The correct eye identification rate is 92.6% during the actual operation.

I. INTRODUCTION

Eye detection is a crucial aspect in many useful applications ranging from face recognition/detection to human computer interface, driver behavior analysis, or compression techniques like MPEG4. By locating the position of the eyes, the gaze can be determined. In this way it is possible to know where people are looking at and understand the behaviors in order to evaluate the interests (for interface purposes) and the attention levels (for safety controls).

A large number of works have been published in the last decade on this subject. Generally the detection of eyes consists of two steps: locating face to extract eye regions and then eye detection from eye window. The face detection problem has been faced up with different approaches: neural network, principal components, independent components, skin color based methods [17, 28]. Each of them imposes some constraints: frontal view, expressionless images, limited variations of light conditions, hairstyle dependence, uniform background, and so on. A very exhaustive review has been presented in [27]. On the other side many works for eye or iris detection assume either that eye windows have been extracted or rough face regions have been already located [16, 18-25].

A careful analysis of the related works suggests some considerations: first of all, the problem of face segmentation, distinguishing faces from a cluttered background, is usually avoided by imaging faces against a uniform background. Second, the common use of skin color information to segment the face region is basically based on cumbersome initializations. Finally, more precise is the location of the eye regions, more reliable are the results of the eye detection algorithms.

To the best of our knowledge, no work has been presented in literature that search directly eyes in whole images, except for active techniques: they exploit the spectral properties of pupil under near IR illumination. In [26] two near infrared multiplexed light sources synchronized with the camera frame rate have been used to generate bright and dark pupil images. Pupils can be detected by using a simple threshold on the difference between the dark and the bright pupil images.

The main objectives of this work is to propose an eyes detection algorithm that is applicable in real time with a standard camera, in a real context such as people driving a car (then with a complex background), and skipping the first segmentation step to extract the face region as commonly done in literature.

The eye detection algorithm works on the whole image, looking for regions that have the geometrical configuration of the edges as the expected ones of the iris. Different iris radius are allowed in order to face with people having different eyes dimensions and also light variations in the distance between the camera and the person. A large number of tests have been carried out on different people, with different eyes colors and dimensions, some of them wearing glasses. The results are surprising good also considering that no constraint has been imposed on the hairstyle and the background. The rest of the paper is organized as follows: Section 2 describes the Parallel Genetic Algorithms in details. Section 3 describes the search process of eyes. The results on different real images are reported in Section 4. Finally, in Section 5 conclusions and future works are presented.

II. GENETIC ALGORITHMS

A sequential GA (Fig. 1) proceeds in an iterative manner by generating new populations of strings from the old ones. Every string is the encoded (binary, real, ...) version of a tentative solution. An evaluation function associates a fitness measure to every string indicating its suitability to the problem. The algorithm applies stochastic operators such as selection, crossover, and mutation on an initially random population in order to compute a whole generation of new strings. Unlike most other optimization techniques, GAs maintain a population of tentative solutions that are competitively manipulated by applying some variation operators to find a global optimum. For nontrivial problems this process might require high computational resources (large memory and search times, for example), and thus a variety of algorithmic issues are being studied to design efficient GAs. With this goal, numerous advances are continuously being achieved by designing new operators, hybrid algorithms, termination criteria, and more [1]. We adopt one such improvement consisting in using parallel GAs (PGAs) and incorporating some advanced heuristics into an overall genetic algorithm.

PGAs are not just parallel versions of sequential genetic algorithms. In fact, they reach the ideal goal of having a parallel algorithm whose behavior is better than the sum of the separate behaviors of its component sub-algorithms, and this is why we directly focus on them. Several arguments justify our work. First of all, PGAs are naturally prone to parallelism since the operations on the representations are relatively independent from each other.Besides that, the whole population can be geographically structured [2,3] to localize competitive selection between subsets, often leading to better algorithms. Evidences of a higher efficiency [4,6], larger diversity maintenance [6], additional availability of memory and CPU, and multisolution capabilities [7] reinforce the importance of the research advances in the field of PGAs.

Using PGAs often leads to superior numerical performance (not only to faster algorithms) even when the algorithms run on a single processor [4,8]. However, the truly interesting observation is that the use of a structured population, either in the form of a set of islands [9] or a diffusion grid [3], is responsible for such numerical benefits. As a consequence, many authors do not use a parallel machine at all to run structured-population models and still get better results than with serial GAs [4,10,11]. Hardware parallelization is an additional way of speeding up the execution of the algorithm, and it can be attained in many ways on a given structured-population GA. Hence, once a structured-population model is defined, it could be implemented in any uniprocessor or parallel machine. There exist many examples of this modern vision of PGAs, namely, a ring of panmictic GAs on a MIMD computer, a grid of individuals on uniprocessor/MIMD/SIMD computers, and many hybrids.

A PGA has the same advantages as a serial GA, consisting of using representations of the problem parameters (and not the parameters themselves), robustness, easy customization for a new problem, and multiple-solution capabilities. These are the characteristics that led GAs and other EAs to be worth of study and use. In addition, a PGA is usually faster, less prone to finding only sub-optimal solutions, and able to cooperate with other search techniques in parallel. For an overview of the applications of PGAs, see [5,10,12,13]. Also, there is a lot of evidence of the higher efficacy and efficiency of PGAs over traditional sequential GAs (e.g., [13–15]).

```
Initialize population
//with randomly generated solutions
Repeat i=1,2,…      //reproductive loop
       Evaluate solutions in the population
       Perform competitive selection
       Apply variation operators
Until convergence criterion satisfied
```

Figure 1. Pseudo-code of a sequential genetic algorithm

ALGORITHMIC DESCRIPTION OF A PGA

In this section, we formalize and visualize the different types of PGAs from a unifying point of view. The outline of a general PGA is shown in Fig. 2. As a stochastic technique, we can distinguish three major steps, namely, *initial sampling*, *optimization*, and checking the *stopping criterion*.

Therefore, it begins (t=0) by randomly creating a population $P(t=0)$ of μ structures, each one encoding the p problem variables on some alphabet. An evaluation function Φ is needed each time a new structure is generated in the algorithm. This evaluation is used to associate a real value to the (decoded) structure indicating its quality as a solution to the problem. Some selected structure encodes tentative solutions to complex systems in a single genetic individual. This individual is used to simulate this complex system every time an evaluation is requested by the algorithm. Consequently, it can be inferred that considerable time is spent when complex and real-life applications are being tackled with GAs, thus supporting our claims about the need of using PGAs as more efficient search methods.

Afterward, the GA iteratively applies some set of variation operators on some selected structures from the current population. The goal is to create a new pool of λ tentative solutions and evaluate them to yield $P(t+1)$ from $P(t)$. This generates a sequence of populations $P(0)$, $P(1)$, $P(2)$, … with increasingly fitter structures. The stopping criterion ι is to fulfill some condition like reaching a given number of function evaluations, finding an optimum (if known), or detecting stagnation in the algorithm after a given number of generations. The selection $S_{\Theta s}$ uses the relationship among the fitness of the structures to create a mating pool. Some parameters Θ_s might be required depending on the kind of selection. Typical variation operators are crossover ($\otimes$ binary operator) and mutation (m, unary operator). Crossover recombines two parents by exchanging string slices to yield two new offspring, while mutation randomly alters the contents of these new structures. They both are stochastic operators whose behavior is governed by a set of parameters like a probability of application: $\Theta_c=\{\rho_c\}$ -high- and $\Theta_m=\{\rho_m\}$ -low-.

```
t=0;
initialize:        $P(0)=\{\vec{a}_1(0),...,\vec{a}_\mu(0)\}\in I^\mu$;
evaluate:          $P(0)=\{\Phi(\vec{a}_1(0)),...,\Phi(\vec{a}_\mu(0))\}$;
while $\iota(P(t))\neq$ true do          //reproductive plan
        select:        $P'(t)=s_{\Theta_s}(P(t))$;
        recombine:     $P''(t)=\otimes_{\Theta_c}(P'(t))$;
        mutate:        $P'''(t)=m_{\Theta_m}(P''(t))$;
        evaluate:      $P'''(t):\{\Phi(\vec{a}_1'''(0)),...,\Phi(\vec{a}_\lambda'''(0))\}$;
        replace:       $P(t+1)=r_{\Theta_r}(P'''(t)\cup Q)$;
        <communication>
        t=t+1;
end while
```

Figure 2. Pseudo-code of a parallel genetic algorithm

Finally, each iteration ends by selecting the μ new individuals that will comprise the new population. For this purpose, the temporary pool $P'''(t)$ plus a set Q are considered. Q might be empty ($Q=\varnothing$) or contain part or all of the old population $Q=P(t)$. This step applies a replacement policy r that uses the temporary pool (and optionally the old pool) to compute the new population of μ individuals. The best structure in one population usually deterministically survives in the next generation, giving rise to the so-called elitist evolution (or R-elitism if $R>1$ -more than one string survives-). The best structure ever found is used as the PGA solution.

III. PROCESS OF EYE DETECTION

The idea behind this study is quite simple: the eyes can be easily located in the image since the iris is always darker than the sclera no matter what color it is. In this way the edge of the iris is relatively easy to detect as the set of points that are disposed on a circle. The first step in applying PGAs to the problem of feature selection for eye detection is to map the pattern space into a representation suitable for genetic search. Since the main interest is in representing the space of all possible subsets of the original feature list, the simplest form for image base representations considers each feature as a binary gene. Each individual chromosome is then represented as a fixed-length binary string standing for some subset of the original feature list. In this method, first of all the pupil and the edge of the eye are extracted, in addition the position of the eyes is detected more accurately.

For the extraction of the eye area, chromosome of individual is set as the first former array composed position information and the size of eye's outlines and pupil. Moreover, fitness of individual is obtained from evaluation function, which pays attention to three features of eyes (difference between white of eye and pupil, color, shape, and size of pupil, edge of eye). The eye area is extracted by chromosome information on the individual with the maximum fitness when evolution completed.

For detection, the pupil is expressed in circle, and the outlines of eye are expressed in the oval, and the pupil is assumed to be at centers of eye. It is defined that the chromosome of individual is composed as X and Y coordinates which shows center of eyes, radius of circle that shows size of pupil and shape of oval which shows outlines of eyes. The first former array of each center coordinates is ten bits, a radius of the circle and oval major axis and minor axis is six bits, total 36 bits, makes. In PGA random initialization is used.

Next, whether the defined chromosome is suitable as the eye is decided according to the evaluation function. Since this method pays attention to the features of eye, "Eyes have white of the eye and pupil", "The shape of the pupil is near circle, and the color is a black", "The outline of eye can be approximated to the oval", the following three are used as an evaluation function.

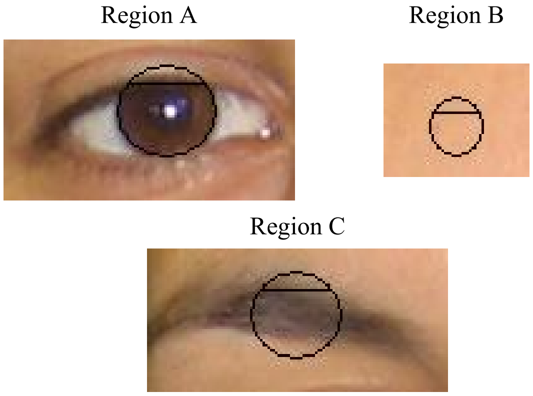
DENSITY DIFFERENCE OF SMALL AREA (AREA SURROUNDING EYE)

Eye has white of the eyes and pupil, it is feature that high density difference between white area and black area. In a word, if there is a big change in the density value of the pixel in a certain area, it is concluded this area is near eyes or the areas around eyes. Then, products of two high-ranking values of the density difference in the area are used as feature. Where eyes are enclosed, the density change is large in the boundary between white of the eye and pupil, the product has a large value.

RATION OF BLACK PIXELS CONTAINED IN CIRCLE (PUPIL)

It is a feature of the pupil that shape is a circular arc, because upper part of pupil is hidden by above eyelid. The color of pupil looks black in the brightness image and the turn of the pupil in enclosed with white of the eye. Therefore, the portion of black pixel in circular arc and the number of white pixels in circular arc surroundings are used as among of features. In Fig. 3, it is evaluated that area A where eyes are shown has high value. But the case of area B (skin) or area C (eyebrow) does not have proper value of white and black pixels in the circular arc, then the evaluation is bad.

Moreover, the gap of the center is evaluated by obtaining the difference of the radius of circle which is in scribe to the pupil and the circle with the string. As shown in Fig. 4, if the circular arc center shifts from the center of the circle, the difference of the radius is large, if the center is near each other, the difference of the radius is small.



Region A

Region B

Region C

Figure 3. Evaluation of Circle

Figure 4. Gap of center

EVALUATION OF OVAL AREA (EDGE OF EYES)

Density values in oval area except area of pupil and the difference of shape between oval and eyes are features, for obtaining fitness. Eye is composed of white of eye and pupil. Therefore, the number of the black pixel in oval area excepted pupil is evaluated. In this feature, few black pixel is better fitness. As shown in Fig. 5, in this case, black pixels are few and the evaluation is good, but the case of eyebrow and hair, is badly evaluated because a lot of black pixel exists. The difference of shape between oval and outline of eye is obtained by using edge image. The sum of distance from 12 points is obtained, top, bottom, right and left. If the sum distance is big value, the oval and the shape pf eyes are greatly different, if the sum is small, those shapes looks like.

IV. EXPERIMENTAL RESULTS

In this section the experimental results obtained on a large number of images are shown. A large number of different persons have been used to take images with different eyes colors, different dimensions and different shapes. No constraints have been imposed on the background. An image can contain more than one person. The proposed technique will be successful for finding all of the eyes. Faces are oriented into various directions and positioned arbitrarily in cluttered background.

The proposed approach can achieve an accuracy rate of about 98.6%, and can detect the eye with various orientations. Some experimental results are shown in Fig. 5. The software has been implemented by using Visual C++ on a Pentium IV 2.6 Ghz without any image preprocessing specialized hardware. A Firewire Camera with a frame rate of 16-25 fps and with a focal length of 6.5 mm has been used to take the images. Although no code optimization has been implemented the processing time seems encouraging: the search of the eyes in an image of 640x480 takes quite 0.24 sec. A tracking procedure could be used to reduce the search time in a sequence of images, once the eyes have been correctly detected. Besides, the introduction of code optimization and specialized hardware for the PGAs will easily produce real time performances for standard camera frame rate.

V. CONCLUSION

In this study, an effective algorithm for eyes detection in arbitrary images is presented. PGA is used to detect the eyes according to the information based on the features of eyes, like shape of the eye and pupil white of eye and pupil. The proposed technique does not impose any constraint on the background and does not require any preprocessing step for face segmentation. Eye is detected without receiving charge of the lighting and effect of face of direction. High detection rates have been obtained. The results are surprisingly good also when the eyes are not completely open.

REFERENCES

1. J.H. Holland: Adaptation in natural and artificial sys.. U. of Michigan Pr., Ann Arbor, 1975.
2. M. Gorges-Schleuter: ASPARAGOS: an asynchronous parallel genetic optimisation strategy. Procs. of the 3rd ICGA. J.D. Schaffer (Ed.) Morgan Kaufmann, (1989).
3. P. Spiessens and B. Manderick: A massively parallel genetic algorithm. Proceedings of the 4th Int. Conf. on Genetic Algorithms. R.K. Belew (Ed.) Morgan Kaufmann, (1991).
4. V.S. Gordon and D. Whitley: Serial and parallel genetic algorithms as function optimizers. Procs. of the 5th ICGA. S. Forrest (Ed.) Morgan Kaufmann, (1993).
5. J. Stender (Ed.): Parallel genetic algorithms: theory and applications. IOS Press, (1993).
6. S. Lin, W.F. Punch, and E.D. Goodman: Coarse-grain parallel genetic algorithms: categorization and new approach. Parallel & distributed processing, (1994).
7. T.C. Belding: The distributed genetic algorithm revisited. Proc. of the 6th ICGA. L.J. Eshelman (Ed.) Morgan Kaufmann, (1995).
8. F. Herrera and M. Lozano: Gradual distributed real-coded genetic algorithms. Technical Report #DECSAI-97-01-03, (1997). (Revised version 1998).
9. R. Tanese: Distributed genetic algorithms. Proc. of 3rd ICGA. J.D. Schaffer (Ed.) (1989).
10. E. Alba, J.F. Aldana, and J.M. Troya: A genetic algorithm for load balancing in parallel query evaluation for deductive relational databases. Procs. of the I. C. on ANNs and GAs. D.W. Pearson, N.C. Steele, and R.F. Albrecht (Eds.) Springer-Verlag, (1995).

Figure 5. Some images with eyes detected

11. E. Alba, J.F. Aldana, and J.M. Troya: Full automatic ANN design: a genetic approach. In: IWANN'93. New Trends in Neural Computation, LNCS 686. Springer-Verlag, (1993).
12. M. Lozano: Application of fuzzy logic based techniques for improving the behavior of GAs with floating point encoding. Ph.D. Thesis. University of Granada, (1996).
13. E. Alba and C. Cotta: Evolution of complex data structures. Informatica y automatica. (1997), 30(3), 42–60.
14. E. Cantu-Paz: A summary of research on parallel genetic algorithms. R. 95007, (1995). Also revised version, IlliGAL R. 97003, (1997).
15. J.L. Ribeiro, *et. al.*: Genetic algorithm programming environments. In: Parallel genetic algorithms: theory & applications. J. Stender (Ed.) IOS Press, (1993).
16. T. Kawaguchi, M. Rizon Iris detection using intensity and edge information, Pat. Rec. 36 (2003) 549-562.
17. H. A. Rowley, S. Baluja, T. Kanade: Neural Network-Based Face Detection IEEE Trans. on Pattern Analysis and Machine Intelligence Vol. 20,No. 1, Jan. 1998, pp 23-38.
18. Y. Li, Xiang-lin Qi, Yun-jiu Wang,: Eye detection by using fuzzy template matching and feature-parameter based judgement Pattern Recognition Letters 22 (2001) 1111-1124.
19. S. Baskan, M. Bulut, V. Atalay: Projection based method for segmentation of human face and its evaluation Pat. Rec. Letters 23 (2002) 1623-1629.
20. S. Sirohey, A. Rosenfiled Eye detection in a face image using linear and nonlinear filters Pattern Recognition 34 (2001) 1367-1391.
21. S. Sirohey, A. Rosenfiled, Z. Duric A method of detection and tracking iris and eyelids in video Pattern Recognition 35 (2002) 1389-1401.
22. M. Rizon, T. Kawaguchi: Automatic extraction of iris with reflection using intensity information. Proc. of the 3th Iasted Conf. on Vis. Imaging and image processing, 2003.
23. P. Smith, M Shah, N. da Vitoria Lobo: Determining Driver Visual Attention with One Camera, IEEE Transactions on Intelligent Transportation Systems.
24. H. S. Ling, L. M. Ang and K. P. Seng: Eye Detection in Face Images, International Conference on Neural Networks and Brain (ICNN&B '05), Oct. 2005.
25. Jiatao Song, Zheru Chi, Zhengyou Wang, and Wei Wang: Locating Human Eyes Using Edge and Intensity Information: Proceedings of ICIC 2005, LNCS 3645, pp. 492 – 501, Springer-Verlag Berlin Heidelberg, 2005.
26. C.H. Morimoto, D. Koons, A. Amir, M. Flickner: Pupil detection and tracking using multiple light sources Image and Vision Comp. 18, (2000) 331-335.
27. M. H. Yang, D. Kriegman, N. Ahuja Detecting Faces in Images: A Survey IEEE Transaction on Pattern An. and Machine Int. Vol. 24, No. 1 January 2002.
28. R. Hsu, M. Mottleb, A. K. Jain Face Detection in Color Images IEEE Transaction on Pattern Anlysis and Machine Intelligence Vol24, No. 5, May 2002.