GECCO 2006
Tutorial

Evolutionary Image Analysis and Signal Processing

Stefano Cagnoni

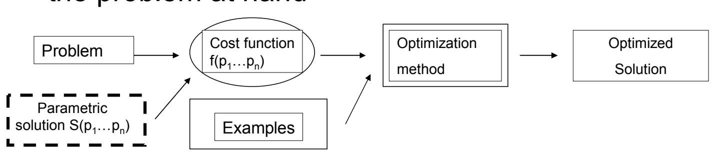
OUTLINE

- **Using Heuristic Parameter Optimization to Solve Problems**
- **The Vision Chain**
- **What to Optimize (and How)**
- **EC Tools**
- **Examples**
- **Discussion**
- **Commercials!**

HEURISTIC PARAMETER OPTIMIZATION AND REAL-WORLD PROBLEMS

The function to be optimized:

- describes parametrically the solution to a practical problem
- is optimized based on performance achieved on a set of examples which are representative of the problem at hand



A DIFFERENT CONCEPT OF “DESIGN”

From:

- ***defining* exact** solutions, **justified by** an underlying **theory**

To:

- ***searching*** solutions which ***work well***, by:
  - defining a quality criterion to measure the effectiveness (cost) of possible solutions
  - choosing a method which maximizes (minimizes) it.

WHEN ?

- No direct solution is available
- Problem specifications can be provided only qualitatively or through examples
- Behaviors or phenomena can be described or measured with little precision (e.g., noisy signals)
- Little a priori knowledge (or none at all!)
- Integration of modules to which any the previous conditions applies

COMPUTER VISION

The “art” of making computers see (and understand what they see)

Sub-topics (the ‘vision chain’)

Image Acquisition
Image Enhancement
Segmentation
3D-Information Recovery
Image Understanding

COMPUTER AND HUMAN VISION

HUMAN	COMPUTER
Perception	Image Acquisition
Selective information extraction	Feature Enhancement (signal/image processing)
Grouping by ‘similarity’	Segmentation
Explicitation of spatial relationships	3D-Information Recovery
Object recognition and semantic reconstruction	Image Understanding

COMPUTER AND HUMAN VISION

HUMAN	COMPUTER
Perception	Image Acquisition
Selective information extraction	Feature Enhancement (signal/image processing)
Grouping by ‘similarity’	Segmentation
Explicitation of spatial relationships	3D-Information Recovery
Object recognition and semantic rec...	Image Understanding

LOW-LEVEL VISION

COMPUTER AND HUMAN VISION

HIGH-LEVEL VISION

HUMAN	COMPUTER
	isition
Selective information extraction	Feature Enhancement (signal/image processing)
Grouping by 'similarity'	Segmentation
Explicitation of spatial relationships	3D-Information Recovery
Object recognition and semantic reconstruction	Image Understanding

APPLICATION FRAMEWORKS

- Optimization of parameters of specific objective functions:
  - related with a well-defined task.
  - for a whole system.
- Generation of solutions *from scratch*
  Optimization of performances based on:
  - Specific objective functions
  - interactive qualitative comparisons between solutions

APPLICATIONS TO SIGNAL/IMAGE PROCESSING AND PATTERN RECOGNITION

- Optimization of filter/detector AND algorithm parameters for event detection and image segmentation.
- Qualitative optimization of image processing algorithms.
- Design of implicitly parallel binary image operators and classifiers.

EC-BASED IMPLEMENTATIONS

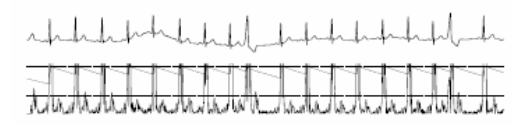
- GA-based design of a QRS detector for ECG signals.
- Optimization of a 3D segmentation algorithm for tomographic images based on an elastic contour model.
- GP-based design of lookup tables for color processing of MR images.
- SmcGP-based low-level image processing and low-resolution character recognition.

FILTER DESIGN/OPTIMIZATION

- Typical problems:
  - event detection
  - image segmentation
- Basic structure of a detection / segmentation algorithm
  - Filter => signal (contrast) enhancement
  - Detector => event (feature) detection

SIGNAL PROCESSING

- Signal Enhancement and Event Detection



- Linear filter:

$$y_i = a_0 + \sum_{k=1}^{N} a_k x_{i-k}$$

EVOLUTIONARY DESIGN OF QRS DETECTORS

Given:

- Filter/detector layout
- Training set
- Fitness function

Optimize:

- Filter coefficients
- Detector threshold
- Other parameters regulating the adaptive behavior of the detector

FILTER LAYOUT

- Linear:

$$y_i = a_0 + \sum_{k=1}^{10} a_k x_{i-k}$$

- Linear with selected samples:

$$y_i = a_0 + \sum_{k=1}^{5} a_k x_{i-d_k}$$

- Quadratic with selected samples:

$$y_i = \underbrace{\sum_{k_1=0}^{2} \sum_{k_2=0}^{2} \sum_{k_3=0}^{2}}_{\sum k_j \leq 2} a_{k_1 k_2 k_3} x_{i-d_1}^{k_1} x_{i-d_2}^{k_2} x_{i-d_3}^{k_3} \qquad d_i \in (1, 10), \;\; d_{i+1} > d_i$$

OTHER PARAMETERS

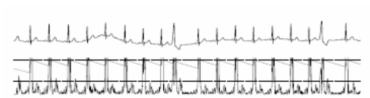
Y_i = adaptive threshold, such that

$$Y_0 = Y_{start}$$
$$Y_i = g((1-\alpha)Y_{i-1} - \beta z_{i-1}(Y_{i-1} - \gamma y_{i-1}))$$

where $g(x) = \max\{Y_{min}, \min[Y_{max}, x]\}$

α = decay rate

β = speed with which Y_i moves towards γy_{i-1}.
γ = percentage of last peak towards which the threshold decays.



EXPERIMENTAL SETUP

TRAINING SET

10 10-second tracts of the ECG from each of the 48 30-minute records of the MIT-BIH Arrhythmia Database (5981 beats out of about 110,000).

FITNESS FUNCTION

$f = f_{max} - (FP^2 + FN^2)$, f_{max} such that $f>0$

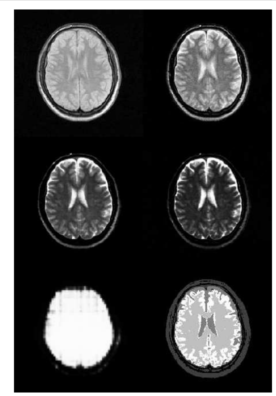
FP = False Positives, FN = False Negatives

RESULTS

- 99.5% average sensitivity (100% on most "normal" recordings)
- Much faster detection with respect to published algorithms yielding comparable results

IMAGE SEGMENTATION

Region based



Contour based

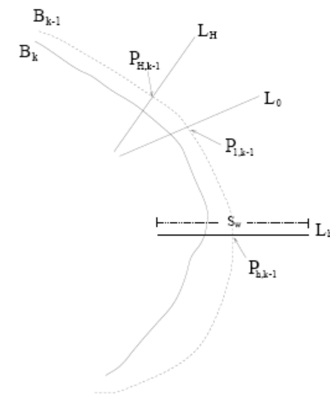
RATIONALE

Pre-defined routines seldom work, so:

- GA optimization of a specific edge detector, via
- interactive specification of a few training contours, followed by
- extraction of the contours of the structure of interest from the whole data set

CONTOUR EXTRACTION

The problem can be reduced to 1D contour detection.



FILTER

c_0, c_k' $c_{k,l}''$ = filter coefficients of 0th, 1st, and 2nd order terms.

d_k = offsets along the scan line

$$o(x_i) = c_0 + \sum_{k=1}^{N_p} (c'_k \cdot x_{i-d_k} + x_{i-d_k} \cdot \sum_{l=k}^{N_p} c''_{k,l} \cdot x_{i-d_l})$$

DETECTOR

2 possible detection schemes:

- Thresholding of filter output

$$O(x_i) = \begin{cases} 0 & \text{if } o(x_i) > T \\ 1 & \text{otherwise} \end{cases}$$

where T = threshold

- Detection of the threshold-crossings of the filter output

$$O(x_i) = \begin{cases} 1 & \text{if } (o(x_i) - Z)\,(o(xi-1) - Z) < 0 \\ 0 & \text{otherwise} \end{cases}$$

where Z= threshold

T and Z are also optimized by the GA

SEGMENTATION

- Definition of a starting contour
- Iterate:
  - – Application of the GA-designed filter to the next contour
  - – Elastic contour model-based interpolation (also optimized by the GA) of the edge points extracted by the filter

TRAINING SET

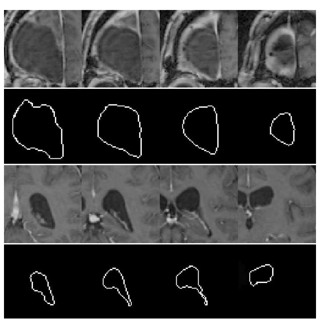
One slice following the one which is used to seed the iterative segmentation process

FITNESS FUNCTION

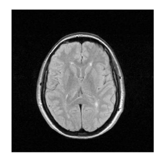
$$F(\{y\}) = K - \sqrt{\sum_{k=1}^{H} d_{y_k}{}^2}$$

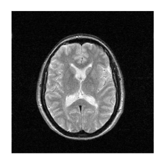
d_{yk} = distance, along scan line L_h, between the actual edge point and the one detected, K = constant

RESULTS



INTERACTIVE EVOLUTION OF LOOKUP TABLES





Aim: given two images $I_1(x,y)$ and $I_2(x,y)$, produce a color image evolved by GP

$I_C(x,y) = F\ (I_1(x,y),\ I_2(x,y)\ )$

with ‘interesting’ features from the point of view of a specific application

Fitness is implicitly defined by the user who acts as the referee of a tournament (of size 2) used in the selection phase

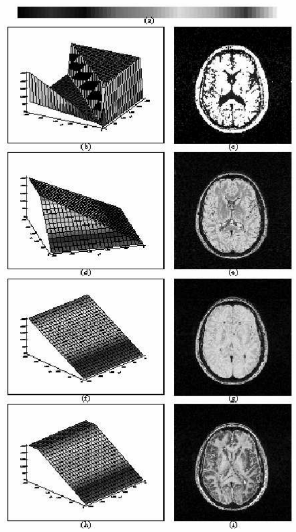
Population size: 15

N. of generations : 10-15

Crossover rate: 0.7

No Mutation

Results of 3 different experiments



SUB-MACHINE CODE GENETIC PROGRAMMING (Poli)

GP variant: programs are evolved which use bitwise logic operations applied to a packed encoding of multiple binary data

Programs are executed on sequential computers but they implicitly implement un a SIMD (Single Instruction Multiple Data) paradigm

Software implementation of recent CPU's multimedia instruction set extensions (es. Intel MMX, AMD 3DNow)

SUB-MACHINE CODE GENETIC PROGRAMMING

Unsigned long variables are used (32 or 64 bit long depending on the compiler or the computing architecture) to encode the binary array of inputs

The bit string may encode consecutive samples of a temporal sequence, a row or a window within an image, etc.

A whole block of data is affected by a single boolean operation (SIMD paradigm)

SUB-MACHINE CODE GENETIC PROGRAMMING

Advantages

• operating in parallel on multiple data makes fitness evaluation more efficient

• fitness can be evaluated on multiple fitness cases at the same time with a single operation sequence

Limitations

• Impossible to apply different operations (or different weights) to data from the same block: the long int variable is an array of independent data which undergo the same operation

AIMS OF THE APPLICATION

- To test SmcGP effectiveness on a real-world problem (car license plate recognition)
- To compare results achieved by the SmcGP-evolved programs with the corresponding algorithms used in the APACHE license-plate recognition system

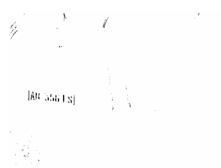
EXPERIMENTAL TEST-BED
(APACHE PLATE-RECOGNITION SYSTEM)

- **Main task**: car license-plate recognition
- **Data**: 130 images of running cars
- **Sub-tasks**: plate extraction and character recognition

PRE-PROCESSING

Input image

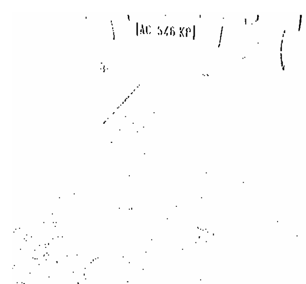
Grey-level image



H.gradient image

The horizontal-gradient image is thresholded to obtain a binary image with only the strongest edges.

APACHE: PLATE EXTRACTION



The plate region is the region where the horizontal edge density is highest

APACHE: CHARACTER RECOGNITION

Characters are extracted from the plate and

an LVQ neural net is used to recognize them

CHARACTER RECOGNITION

Recognition of digits represented by binary two-dimensional patterns: 10 specialized binary classifiers have the pattern as input and produce as output:

1 if the patterns belongs to the class corresponding to the classifier

0 if the pattern belongs to another class

CLASSIFICATION BY INDEPENDENT CLASSIFIERS

Advantages

- Each classifier is specialized and yields high performances
- Classifiers are very ‘compact’: they don’t need to consider features belonging to several classes

Disadvantages

Possible ambiguous classifications:

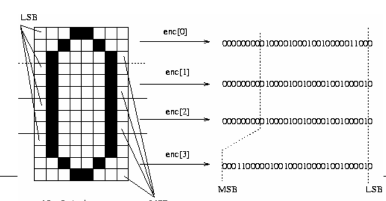
- The output of all classifiers is 0
- The output of more than one classifier is 1

A disambiguation mechanism is needed

INPUT ENCODING (TERMINAL SET)

Input Pattern: binary digits of size 13x8
104 bits may be represented using 4 unsigned long variables.

72 bits of the pattern are packed into the 24 least significant bits of the first 3 long int variables
The remaining 32 are packed into the fourth one



FUNCTION SET + ERCs

Binary bitwise operators: AND OR NOT XOR

Circular shift operators: SHR, SHR2, SHR4,
SHL, SHL2, SHL4

Ephemeral Random Constants (ERC):
32 bit unsigned long

FITNESS FUNCTION

2 fitness functions have been considered

FIT1= number of correct classifications (TP+TN)

FIT2= sqrt (Sensitivity ^{2} + Specificity ^{2})

NB If training data are uniformly distributed, then the negative case shown to each classifier are 9 times as many as the positive ones

=> FIT1 privileges specificity
FIT2 keeps better balance between the two properties

EVOLUTION PARAMETERS

Population : 1000
Survival rate : 17 %
Crossover rate: 80%
Mutation rate : 3%

Tournament selection with tournament size = 7

300 to 2000 iterations

TEST SET

Database of plate digits collected at toll booths of Italian highways

about 11000 digits of size 13x8 from real plates binarized with threshold=0.5 (0=black; 1 = white)

6024 in the training set

5010 in the test set (exactly 501 per class)

CLASSIFIERS

Set of 10 binary classifiers (one for each class)

For each classifier:

Input : unsigned long pattern[4]

Output : unsigned long out

Each classifier actually produces 32 independent binary outputs: the output bit which has yielded the highest fitness on the training set is taken as the actual output of the classifier

CLASSIFICATION RULE

The same pattern is input into all classifiers.

10 outputs are produced of which one (hopefully) is 1 and the others are 0.

If the output is ambiguous an external tie-breaker is applied (the LVQ classifier embedded in the APACHE system).

For all-0 outputs this is the only possible remedy. In the case of more than one active outputs it is possible to train a second classifier set to act as tie-breakers.

FITNESS FUNCTION

FIT1 = TP+TN has been used.

- High specificity => high positive predictivity
- All-zero classifications are less than 5%
- Ambiguous cases less than 1 %

About 95% of the digits is directly classified by the basic classifier set (accuracy: training 99.98%, testing 98.7%).

The remaining 5% is classified by the LVQ classifier.

RESULTS

Training set

99.65% correct classifications

(99.98% of the correctly classified 99%)

Test Set

97.43% correct classifications

(98.7% of the correctly classified 95.3%)

About 0.1 microseconds per classifier (1 microsecond per pattern) on a Pentium IV 3 GHz computer after compiling the resulting programs.

IMPLEMENTATION: PREFIX NOTATION

```
(AND (AND (NOT (SHR2 (OR PAT3 PAT2)))
          (SHL4 (SHL4 (NOR (SHR2 PAT2)
                           (AND PAT3
                                (NOR (SHR4 (SHL PAT1)) PAT1))))))
      (SHL4 (SHL4 (NOR (SHL2 (SHL4 PAT3))
                       (OR (OR (SHR PAT3)
                               (AND (NOR PAT2
                                         (SHL2 PAT3))
                                    (SHR2 (SHR4 (NOT (SHR PAT3))))))
                           (AND (SHR PAT2)
                                (SHR2 (SHR4 PAT3)))))))
```

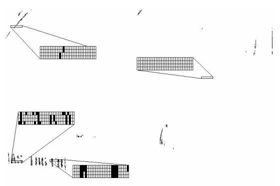
INFIX NOTATION AND C SOURCE

```
( ( ( NOT ( SHR2 ( ( PAT3 ) OR  ( PAT2 ) ) ) ) ) AND ( SHL4 ( SHL4 ( NOT
( ( SHR2 ( PAT2 ) ) OR (( PAT3 ) AND ( NOT ( ( SHR4 ( SHL ( PAT1 ) ) )
OR ( PAT1 ) ) ) ) ) ) ) ) ) AND ( SHL4 ( SHL4 ( NOT ( (SHL2 ( SHL4 (
PAT3 ) ) ) OR ( ( ( SHR ( PAT3 ) ) OR ( ( NOT ( ( PAT2 ) OR ( SHL2 (
PAT3 ) ) ) ) ) AND ( SHR2 ( SHR4 ( NOT ( SHR ( PAT3 ) ) ) ) ) ) ) OR ( (
SHR ( PAT2 ) ) AND ( SHR2 ( SHR4
( PAT3 ) ) ) ) ) ) ) ) ) )

unsigned long class0 (unsigned long p1, unsigned long p2,
                      unsigned long p3, unsigned long p4)
{
 return ( ( ( ~ ( SHR2 ( ( p3 ) | ( p2 ) ) ) ) ) & ( SHL4 ( SHL4 ( ~ (
( SHR2 ( p2 ) ) | ( ( p3 ) & ( ~ ( ( SHR4 ( SHL ( p1 ) ) )| ( p1 ) ) )
) ) ) ) ) ) & ( SHL4 ( SHL4 ( ~ ( ( SHL2 ( SHL4 ( p3 ) ) ) | ( ( ( SHR
( p3 ) ) | ( ( ~ ( ( p2 ) | ( SHL2 ( p3 ) ) ) ) ) & ( SHR2 ( SHR4 ( ~
( SHR ( p3 ) ) ) ) ) ) ) ) | ( ( SHR ( p2 ) ) & ( SHR2 ( SHR4 ( p3 ) ) )
) ) ) ) ) ) );
}
```

SmcGP-BASED PLATE DETECTION

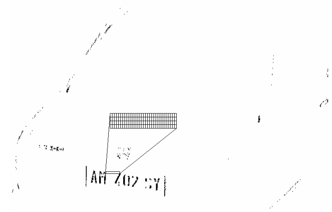
- Input data: 4 unsigned long words encoding a window, of size 32x4 pixels, from the binarized gradient image
- Desired output: 1 if the window belongs to the plate
  0 otherwise



TRAINING SET

- 80/130 images
- Input data: gradient image
- 100 samples/image: 60 from the plate area, 30 from around the plate, 10 from anywhere else in the image
- 366 negative + 4824 positive = 5190 training samples

Empty samples (that can be found both inside and outside the plate) have been purged.



EVOLUTION PARAMETERS

- Population = 1000
- Tournament selection of size = 7
- 80% crossover
  17% survival
  3% mutation

Fitness function

$$F' = \sqrt{\frac{\left(\frac{(n_{Neg}-FP)^2}{n_{Neg}^2} + \frac{(n_{Pos}-FN)^2}{n_{Pos}^2}\right)}{2}} = \sqrt{\frac{Specificity^2 + Sensitivity^2}{2}}$$

$$F'' = Program\_size / 10000000$$
$$F = F' + F''$$

RESULTS (TRAINING SET)

Performance of the best program:

Specificity = 323/366 (88.25%)

Sensitivity = 4045/4824 (83.85%)

Program size : 1087

Fitness: 0.142296

TYPICAL RESULTS

The same algorithm that APACHE directly applies to the gradient image can be applied to this image to improve plate localization

RESULTS

- When several edges are present outside the plate, the 'basic' algorithm may fail.
- 'Filling the gaps' in the plate increases robustness
- The evolved algorithm produces a 10-fold increase in the density of detected pixels within the plate.

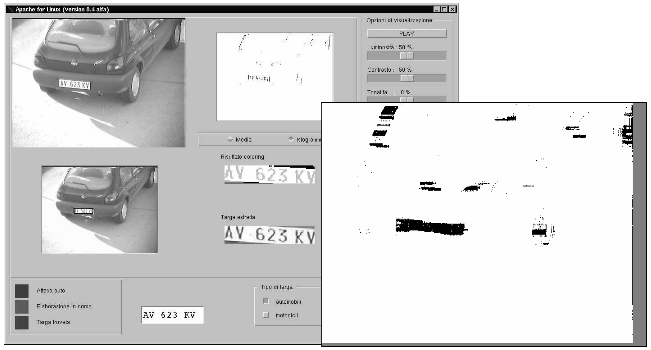
	GRADIENT (%)	GENETIC (%)	PLATE PIXELS
AVERAGE	367 (6.48%)	3513 (62.05%)	5661

- The increase in the number of false detections is limited and mostly harmless.

	GRADIENT (%)	GENETIC (%)	BACKGROUND PIXELS
AVERAGE	470 (0.12%)	8157 (2.00%)	407758

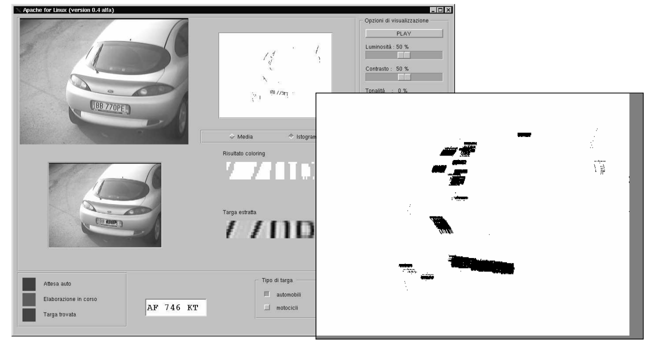
COMPARISON WITH APACHE

- Both algorithms detect the plate correctly



COMPARISON WITH APACHE

- APACHE fails, the genetic program detects the plate



OTHER RESULTS

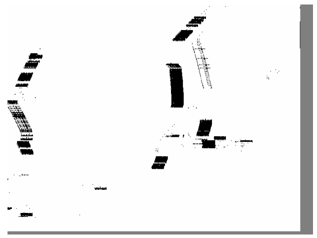
- Quasi-ideal results

OTHER RESULTS

- The plate is detected, in spite of a lot of noise

OTHER RESULTS

- No edges are detected in the plate region: failure



SmcGP RESULTS

- Digit recognition:
  - Performance close to the LVQ classifier with a 10-fold reduction of computation time
- Plate detection:
  - Improved accuracy with a limited increase of computation time
  - The computation efficiency of SmcGP classifiers limits the effects of the overhead added by the GP-evolved stage

CONCLUSIONS

In many cases in which a signal/image processing/understanding problem can be reformulated as an optimization problem, evolutionary computation provides powerful and effective tools to search for 'good' solutions.

BIBLIOGRAPHY

- Poli R., Cagnoni S., Valli G., ``Genetic design of optimum linear and non-linear QRS detectors'', *IEEE Trans. on Biomedical Engineering* , 42(11):1137-41,1995.
- Poli R., Cagnoni S. ``Genetic programming with user-driven selection: experiments on the evolution of algorithms for image enhancement'', in Koza J.R., Deb K., Dorigo M., Fogel D.B., Garzon M., Iba H., and Riolo R.L. (editors), *Genetic Programming 1997: Proceedings of the Second Annual Conference*, July 13-16, 1997, Stanford University. San Francisco, CA: Morgan Kaufmann, pp. 269-277, 1997.
- Cagnoni S., Dobrzeniecki A.B., Poli R., Yanch J.C. ``Genetic-algorithm-based interactive segmentation of 3D medical images'', *Image and Vision Computing Journal*, 17(12):881-896, 1999.
- Cagnoni S., Bergenti F., Mordonini M., Adorni G., "Evolving binary classifiers through parallel computation of multiple fitness cases". *IEEE Transactions on Systems, Man and Cybernetics Part B-Cybernetics*, 35(3):548-555, 2005.

UPCOMING BOOK

Genetic and Evolutionary Image Processing and Computer Vision

EURASIP Book Series in Signal Processing

Editors: S. Cagnoni, E. Lutton, G. Olague

UPCOMING WORKSHOP

EvoIASP 2007: Ninth European Workshop on Evolutionary Computation for Image Analysis and Signal Processing
(as part of EvoWorkshops 2007)

Valencia, Spain
April 2007

Submission Deadline: Early November, 2006