Multi-Genomic Curve Extraction

Raphaël Labayrade
Université de Lyon, 69003 Lyon, France
ENTPE, LGCB, 3 rue Maurice Audin
69120 Vaulx-en-Velin, France
raphael.labayrade@entpe.fr

Mathias Ngo
Université de Lyon, 69003 Lyon, France
ENTPE, LGCB, 3 rue Maurice Audin
69120 Vaulx-en-Velin, France
mathias.ngo@entpe.fr

Abstract

We present Multi-Genomic Curve Extraction (MGCE), a robust method to extract curves in noisy datasets and images. Unlike other robust extraction methods, MGCE does not require to choose the global curve model to extract prior to the process. Instead, it identifies the inliers with respect to an underlying set of local models which number and associated data subsets are automatically determined during the run of the algorithm. As MGCE attempts to minimize this number, the robustness of the inlier extraction is reinforced. The method relies on Multi-Genomic Algorithms (MGA) which are an extension of Genetic Algorithms (GA) designed to handle populations of solutions with variable-length chromosomes. Numerical experiments provide insights about the performance of the method and its applicability to road lane border detection.

1 Introduction

Extracting models in images is a key problem in image processing. It is often difficult since the data extracted from images are usually sparse and noisy, raising the problem of robust model extraction: the model should be adjusted to inliers only while outliers should be rejected. At the same time the adjusted model should perform interpolation in areas where there is no relevant data. The problem becomes more difficult and nearly inextricable if no prior knowledge is available about the model to extract – which is often the case in practice. At best, a few assumptions can be reasonably formalized such as continuity and/or smoothness, but the global shape to look for is usually unknown.
Least Squares [1] is a well-known curve fitting technic which is sensitive to outliers. By introducing assumptions about the noise affecting the data, M-estimators [2] can perform robust curve extraction. In the image processing community, an alternative popular method is the Hough transform [3]. The complexity of the algorithm increases with the number of degrees of freedom of the model that is looked for, limiting its practical use to simple models. Another widely used method is RANSAC [4], that randomly samples the dataset and estimates the model parameters from the sampled data subset; the process is iterated several times. The identified model is the one best adjusted to the whole set of data over the iterative process. Genetic Algorithms (GA) have also been used for feature extraction in a image [5]. Former work concluded that while their implementation is different, the robustness of the method is similar to the Hough transform. Historically, the use of GA was rather marginal in the image processing community but they are becoming more popular.
All the technics mentioned above require choosing the model to extract, since their outputs are estimates of the values of the model parameters. Thus, such methods – at least their basic implementation – are not well suited to look for a curve of unknown model.
In this paper, we introduce Multi-Genomic Curve Extraction (MGCE), which aims at identifying the inliers with respect to an underlying curve of unknown model, from a dataset affected by noise and outliers. The method relies on Multi-Genomic Algorithms (MGA) [6] which are an extension of GA designed to handle populations of solutions with variable-length chromosomes.
The remainder of the paper is structured as follows. Section 2 presents the general flowchart of MGA. Section 3 details MGCE. Section 4 is dedicated to numerical experiments. Section 5 concludes.

2 Multi-Genomic Algorithms (MGA)

MGA [6] are a class of optimization algorithms extending GA, that have been recently applied in the context of geometric problems and building design. Contrary to GA, MGA handle different models in the same population of solutions.
In GA, all the solutions are instances of the same model and are encoded as chromosomes of the same length. From an initial random population, crossover is performed between couples of individuals, that creates children. Mutation is also performed, that slightly modifies the individuals. The resulting individuals are then sorted according to their fitness, representing their suitability to the problem. The best individuals are kept for the next generation while the other ones are discarded. The best individual of the last generation is considered to be the solution of the problem.
In MGA, the population can be composed of solutions being instances of different models. Therefore, individuals are instances of different chromosomes; the structure and length of the chromosome of an individual, and thus his genome, can differ from the one of another individual. As multiple genomes are coexisting, the population is multi-genomic: it is handled by a MGA. In addition to classical GA operators, MGA introduces gene insertion and gene deletion, allowing to modify the length of the chromosome of an individual. Moreover, the crossover can be performed between individuals whose genomes are different; thus, a hybrid crossover operator is used instead of the classical GA crossover operator. The general flowchart of MGA is presented in Algorithm 1. It should be noticed the different operators can be implemented in various ways, and are usually problem-dependent.
With respect to GA, MGA present some interesting advantages and emerging properties. First, since va-

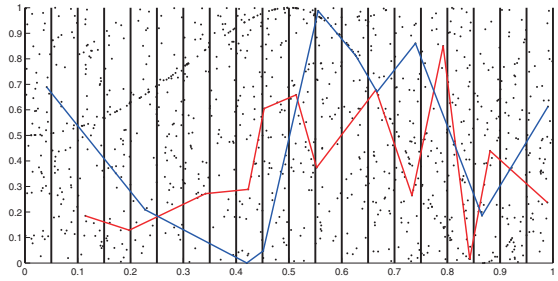

FIG. 1. Vertical areas segmenting the cloud of points, and two individuals in the early generations featuring different numbers of local models.

rious models are handled at the same time, MGA are better suited to problems where the choice of a model is not obvious or trivial. Second, the best model fitted to the problem is automatically identified at the end of the algorithm. Third, the computing time is reduced with respect to a GA handling the model featuring the longest chromosome (i.e. the most general model). For these reasons, MGA seem to be well suited for performing robust extraction of a curve of unknown model.

Algorithm 1: MGA Flowchart

```
Data: Models, num_gen, num_ind
Result: Optimal Solution (MCO)
P = initial_Multi-Genomic_population ;
for i≤num_generations do
    P.assess_fitness() ;
    New_P = [] ;
    while len(New_P)<num_ind do
        parents = Selection(P) ;
        children = Hybrid_Crossover(parents) ;
        New_P.append(children) ;
    P.append(new_P) ;
    P.mutate() ;
    P.gene_insertion() ;
    P.gene_deletion() ;
Return P ;
```

3 Proposed approach: MGCE

3.1 Problem formulation

We suppose a curve of the form $y = f(x)$ (f is unknown) and a cloud of points P that contains n_{in} noisy inliers $(x, f(x_i) + \varepsilon_i)$ where ε_i is a positive or negative amount of noise affecting point i, and n_{out} outliers (xO_i, yO_i) with possibly $n_{out} > n_{in}$ (see Figure 1).
We look for the inliers with respect to a set of local models M_i that approximates the curve f in the cloud of points P. Each model M_i is defined on an interval $I_i = [xb_i, xe_i]$ such as $xb_{i+1} = xe_i$, meaning that the successive intervals are not overlapping. Nor the width of each interval neither the number of local models are set *a priori*. Only the maximum number of models N_{Max} is fixed. In the following implementation, we will consider each model is linear, meaning a piecewise linear approximation is used.

3.2 Multi-Genomic Population

Each individual in the population represent a certain number of consecutive segments that can be different than in the other individuals, resulting in a multi-genomic population.
The cloud of points P is divided in $N_{Max} + 1$ vertical areas (not necessarily of same width), so that each point P belongs to an area (see Figure 1). For implementation purposes, each point is indexed so that the set of points of P belonging to a vertical area is defined as a set of indexes.
An individual of the multi-genomic population is encoded as follows: each segment is defined between two points of P belonging to two different areas – not necessarily consecutive. Nevertheless, two consecutive segments share an extremity point. The chromosome of an individual encodes the list of the indexes of the successive points of P defining the successive segments. The length of a chromosome is thus $N_{ind} + 1$ where N_{ind} is the number of segments of the individual *ind*.

3.3 Initial Population

Initially, for each individual in the population, the number of local linear models N_{ind} is set randomly between 1 and N_{Max}. Then, $N_{ind}+1$ areas are chosen randomly. Then, in each of the $N_{ind}+1$ vertical areas, one point is chosen randomly. A linear local model of the individual *ind* is defined as a segment between the chosen point in one vertical area and the chosen point in the next vertical area.

3.4 Fitness

The fitness is assessed by counting the number of points of the cloud P that are closest from the piecewise linear curve than a distance d. Considering a point (x, y) of P, the distance is the absolute difference between y and the y-coordinate of the local model at x.

3.5 Hybrid One-Point Crossover

The crossover operator can be applied between two parents Par_1 and Par_2 representing different numbers of local models. The crossover point is defined randomly as the right extremity point of a segment of Par_1. At this point, Par_1 is split in two. Par_2 is also split in two, at the left extremity point of the segment closest to the crossover point, that belongs to a vertical area which index is strictly superior to the one of the crossover point. Two children are obtained by merging the split parts of Par_1 and Par_2.

TAB. 1. Point cloud and MGCE parameters.

♯ P	2000	σ	0.01
♯ individuals	50	N_{Max}	20
♯ max gens	10000	t_{local}	68% $\frac{\sharp inliers}{N_{Max}+1}$
a_s	30°	d	2σ

TAB. 2. Detection of linear curve.

% outliers	50%	80%	90%	93%
♯ runs	1000	1000	1000	450
< ♯ gens>	2.82	6.80	10.64	28.30
min	2	2	3	3
max	13	27	49	626
StD	1.18	2.79	4.39	8.97
< ♯ models>	10.84	9.28	9.23	8.97
StD	2.41	2.61	2.62	3.15

TAB. 3. Detection of sin curve.

% outliers	50%	80%	90%	93%
♯ runs	1000	1000	1000	687
< ♯ gens>	3.76	8.69	20.36	28.20
min	2	2	5	9
max	10	21	57	82
StD	1.46	2.92	7.04	10.25
< ♯ models>	13.93	14.25	15.08	16.78
StD	1.76	2.04	1.83	1.51

TAB. 4. Detection of parabolic curve.

% outliers	50%	80%	90%	93%
♯ runs	1000	1000	1000	1000
< ♯ gens>	2.62	6.42	14.85	28.31
min	2	2	2	3
max	7	26	104	2059
StD	0.85	2.49	8.68	70.59
< ♯ models>	11.07	9.97	11.08	11.15
StD	2.57	2.72	2.26	2.04

TAB. 5. Detection of $y = sin(x^2)$ curve.

% outliers	50%	80%	90%	93%
♯ runs	1000	1000	1000	238
< ♯ gens>	4.73	10.68	40.10	108.28
min	2	2	3	6
max	58	89	321	1288
StD	3.33	5.90	30.81	653.10
< ♯ models>	14.63	14.96	17.83	18.14
StD	2.24	2.34	1.69	1.44

3.6 Mutation

The mutation of an individual consists, for one extremity point of one segment chosen randomly, in choosing randomly another point that belongs to the same vertical area. For the individual *ind*, the mutation is performed with the probability $1 - 1/N_{ind}$.

3.7 Gene deletion

The gene deletion operator consists in removing one segment to an individual. To do so, a vertical area where a segment extremity is defined is chosen randomly. The two segments defined from this extremity point are fused into a single one by removing the chosen extremity point. The gene deletion is performed with the probability $1 - 1/N_{ind}$ in the event where:

- the two local models are aligned: $\mid angle\_segments - 180^{\circ} \mid < a_s$ with a_s an arbitrary threshold;
- or the two local models define a peak: $\mid angle\_segments \mid < a_s$. This occurs when the curve is not smooth; a_s can thus be set to guide the search towards more or less smooth curves.

3.8 Gene insertion

The gene insertion operator consists in adding one segment to an individual. To do so, a vertical area where no segment extremity is defined is chosen randomly. Then, a point of the cloud P in this area is chosen randomly. The segment passing through this area is split in two at the level of this point, resulting in two segments. For the individual *ind*, the gene insertion is performed with the probability $1 - 1/N_{ind}$ only if the fitness of the segment is below a threshold t_{local} (which indicates the local model may not be well adjusted to the data before the gene insertion).

4 Numerical experiments

4.1 Curve extraction in noisy cloud of points

In order to assess how MGCE performs, we first created clouds of noisy curve points (straight line, parabola, trigonometric curves). For each curve, a cloud containing 2000 points was created and scaled in the $[0; 1]$ range. Uniform noise between $[-\sigma; \sigma]$ was added to the y-coordinate of the inliers (see Table 1). Uniformly distributed outliers were added to the cloud. Table 1 indicates the values of the MGCE algorithm parameters. Figure 2 presents examples of the curves extracted. Tables 2-5 indicates how, over a large number of runs, the algorithm performs for each extraction and for different amounts of outliers. In all the cases, MGCE succeeded in extracting the inliers of the curve. The required average number of generations < ♯ gens> for convergence increases when the inliers follow a less smooth curve and when the amount of outliers increases. With a population of 50 individuals, this average number ranges from 2.62 to 108.28 meaning the number of average required evaluations ranges from 131 to 5414. This suggests the complexity of the algorithm is low. Unsurprisingly, the average number of local models of the identified solution increases when the curve to extract becomes less smooth.

4.2 Curve extraction in images

In order to assess the potential of MGCE to extract inliers with respect to an underlying curve in images, it was tested on road images. The test images were proposed in [7]. The feature points were obtained by a local threshold and symmetrical horizontal gradient detector. Figure 3 presents examples of results and evidences the ability of MGCE to detect road lane borders of different shapes (linear, curved, "S"-shape) ; heuristics about the fitness assessment (to promote smooth

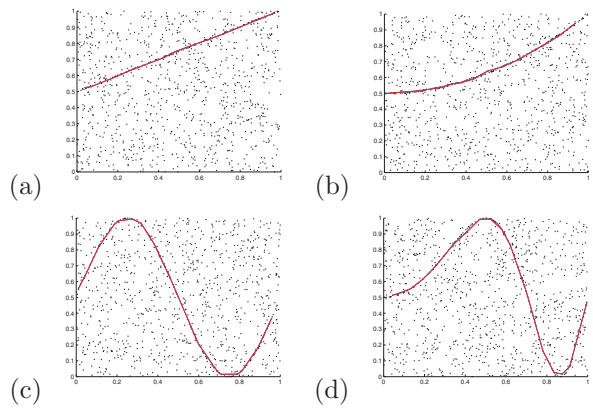

FIG. 2. Examples of curve extraction in noisy point clouds (93% of outliers). (a) linear ; (b) parabolic ; (c) sin ; (d) $y = sin(x^2)$.

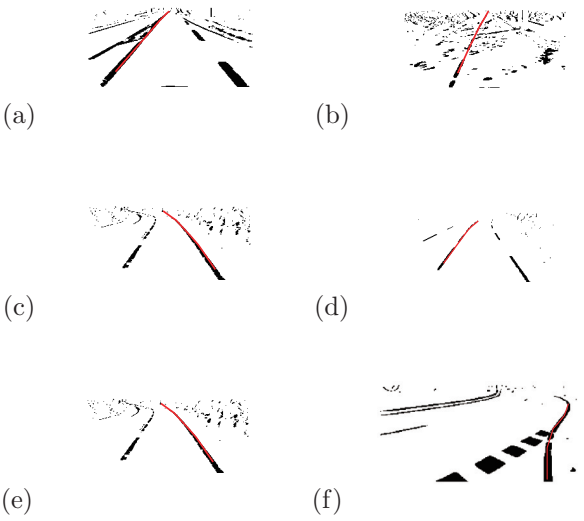

FIG. 3. Examples of lane border detection from feature points in images. (a-b) linear shape ; (c-d-e) curved shape ; (f) "S"-shape.

curves and/or containing a low number of local models) are useful in such an application. Only one lane border is extracted since only one curved is looked for.

4.3 Discussion

The results shown in the previous sections are promising, both in terms of results of inlier extraction with respect to a curve of unknown model, and in terms of complexity. The results could certainly be improved further in the context of specific applications, through fine tuning of the parameters of the algorithms. With a Matlab prototype that could be further refined, the computing time ranges from fractions of a second to a few minutes, suggesting real-time performance is probably reachable with multi-core and/or GPU optimized implementations. Moreover, many applications implement tracking over time, and it is clear that in such a case the computing time to update the result would be significantly lower than for a *from scatch* search. The robustness of MGCE is likely to be provided by its ability to manage sets of different number of local models in the same optimization run, and to dynamically identify how many are needed : only a few are used if it is enough ; others are added automatically if needed. We believe this emerging property inherited from MGA is a key advantage with respect to approaches relying on regular GA. Moreover, since less models are needed than in GA, the decision space is likely to be explored more efficiently – and the computing time reduced. Comparisons with respect to traditional robust extraction algorithms (e.g. RANSAC, M-estimator, Hough transform) on common datasets would be interesting to draw firmer conclusions.

5 Conclusion

We introduced MGCE, a method to identify the inliers with respect to an underlying curve of unknown model, from a dataset affected by noise and outliers. This method is based on MGA that extend GA; its principle relies on the identification of the suitable number of local models that are automatically adjusted to the relevant areas of the dataset. Numerical experiments highlight promising overall performance. Future work will investigate the use of curved local models (e.g. polynomial, splines) instead of the linear one. Also, a generalization of MGCE could be elaborated, aimed at extracting any kinds of patterns – and not only curves of the form $y = f(x)$.

Aknowledgments

The authors are grateful for the financial support of Université de Lyon through the Program "Investissements d'Avenir" (ANR-11-IDEX-0007).

References

[1] Charnes, A., Frome, E. L. and Yu, P. L. (1976). The Equivalence of Generalized Least Squares and Maximum Likelihood Estimates in the Exponential Family. *J. of the American Statistical Association*, 71, 353-169.

[2] Hoaglin, David C. ; Frederick Mosteller and John W. Tukey (1983). *Understanding Robust and Exploratory Data Analysis*. Hoboken, NJ : John Wiley Sons Inc. ISBN 0471097772.

[3] Duda, R. O. and P. E. Hart (1972). Use of the Hough Transformation to Detect Lines and Curves in Pictures. *Communications of the ACM, 15*, 11-15.

[4] Fischler, M. A. and Bolles, R. C. (1981). Random Sample Consensus : a Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM, 24(6)*, 381-395.

[5] Roth, G., and Levine, M. D. (1994). Geometric Primitive Extraction Using a Genetic Algorithm. *Pattern Analysis and Machine Intelligence*, 16(9), 901-905.

[6] Ngo, M. and Labayrade, R. (2014). Multi-Genomic Algorithms. *Proceedings of IEEE Symposium Series on Computational Intelligence*, Orlando, USA.

[7] Veit, T., Tarel, J. P., Nicolle, P., and Charbonnier, P. (2008). Evaluation of Road Marking Feature Extraction. *Proceedings of IEEE Intelligent Transportation Systems Conference*, 174-181.