Evolutionary Lines for Flow Visualization

Wito Engelke, Ingrid Hotz

Department of Science and Technology, Linköping University, Sweden

Abstract

In this work we explore evolutionary algorithms for selected a visualization application. We demonstrate its potential using an example from flow visualization showing promising first results. Evolutionary algorithms, as guided search approach, find close-to-optimal solutions with respect to some fitness function in an iterative process using biologically motivated mechanisms like selection, mutation and recombination. As such, they provide a powerful alternative to filtering methods commonly used in visualization where the space of possible candidates is densely sampled in a pre-processing step from which the best candidates are selected and visualized. This approach however tends to be increasingly inefficient with growing data size or expensive candidate computations resulting in large pre-processing times. We present an evolutionary algorithm for the problem of streamline selection to highlight features of interest in flow data. Our approach directly optimizes the solution candidates with respect to a user selected fitness function requiring significantly less computations. At the same time the problem of possible under-sampling is solved since we are not tied to a preset resolution. We demonstrate our approach on the well-known flow around an obstacle as case with a two-dimensional search space. The blood flow in an aneurysm serves as an example with a three-dimensional search space. For both, the achieved results are comparable to line filtering approaches with much less line computations.

Categories and Subject Descriptors (according to ACM CCS): Visualization [Human-centered computing]: Visualization Techniques—Scientific Visualization

1. Introduction

An evolutionary algorithm is a guided search for an optimum which is based on the principle of natural selection by Charles Darwin [Dar59]. It can be used for optimization problems, where an analytic solution does not exist and an exhausting exploration is not feasible. Thereby, the quality of a small set of candidate solutions is iteratively improved using mechanisms like selection, mutation and recombination. Such algorithms have been used in many applications where it is sufficient to obtain a good, but not necessarily optimal solution in a reasonable time. Examples range from parameter optimization (e.g., curvature of pipes) [BS93], packing, routing and scheduling problems to biological modeling. More examples and in-depth description of building blocks can be found in [Bäc96], a recent book [KBB*16] discusses evolutionary algorithms in the context of computational intelligence. Surprisingly, evolutionary algorithms have not yet found their way into visualization where many methods can be formulated as optimization problem where finding the explicit optimum is not necessarily required.

Geometry-based techniques are one of the most frequently used methods for flow exploration and analysis [MLP*10]. When focusing on lines, numerical integration is used to obtain integral curves i.e., streamlines and path lines. A challenge in this context is to find the most expressive lines to represent the data, highlighting features of interest while avoiding clutter. A typically applied approach is the computation of a high number of candidate lines which are subsequently filtered according to user specified properties.

The filtering can be classified into two categories. First, *explicit filtering* which explicitly removes lines from the set not fulfilling the selected criterion, e.g. directly defined by line predicates [SGSM08] or in a view dependent criterion [LMSC11]. Second, *implicit filtering* which facilitates rendering techniques to emphasize lines of interest while keeping all lines. One example is the work by Günther et al. [GRT13, GRT14] who optimize line transparency with respect to a specific importance criterion. Both approaches share the disadvantage that they discard or hide a high percentage of the calculated lines. This is especially unfavorable in cases where line integration is expensive. Additionally, even a dense seeding does not guarantee finding all interesting features.

This targeted goal of generating the most expressive lines can be understood as an optimization problem and suggests the application of an evolutionary algorithm which we call *Evolutionary Lines*. As a proof of concept we demonstrate results for the case of streamlines in steady data-sets. The main contributions are

- Introduction of evolutionary algorithms to visualization and demonstration of their usefulness for flow visualization.
- Presentation of an evolutionary algorithm for optimizing seed positions for line integration in flow data-sets.
- Discussion of the potential of evolutionary algorithms for other future applications.

DOI: 10.2312/eurovisshort.20181070

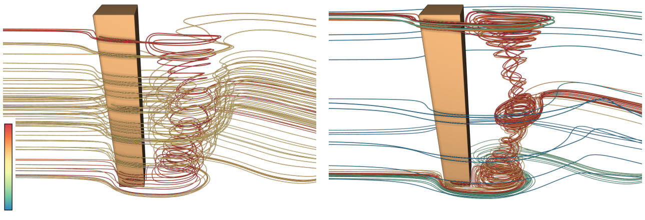


Figure 1: *(a) Explicitly filtered line set. The original set was generated by exhaustive exploration of* Ω *by integration of 3185 points located on a two-dimensional sub-domain. After filtering 70 lines where kept. (b) Result of our evolutionary algorithm after 12 iterations with* $n_p = 100$ *and 70 best shown. For both, filtering (a) and the evolutionary algorithm (b) the same criterion was used.*

2. Evolutionary Lines

Evolutionary algorithms are designed to solve an optimization problem of the following form.

$$\mathscr{H} = \{x \in \Omega | \forall x' \in \Omega : f(x) \succeq f(x')\} \tag{1}$$

The aim is to find an element x in the search space Ω which optimizes a function $f : \Omega \to \mathbb{R}$ in Ω. The fitness function f defines a quality measure for the candidate solutions and a comparison operator $\succeq$ in Ω. Based on this measure a set of initial random candidates is iteratively improved. The main mechanisms used are: *Selection* - keeping the best candidates; *CrossOver* - combining the genome of candidates to generate new offsprings; *Mutation* - random altering of a candidate. The specific implementation of the building blocks of evolutionary algorithms are highly problem specific. Additionally, parameters e.g., population size, mutation probability steer the algorithm.

In the following we consider the visualization task of generating an expressive set of streamlines for a steady flow data-set with respect to a given criterion. This goal is comparable with explicit streamline filtering. In this case the search space Ω is defined by the streamlines' seeding domain. The fitness function is defined as predicate, which can be obtained from the line geometry (e.g., line length, line curvature, etc), or directly from the flow data-set (e.g., velocity magnitude, vorticity, λ_2 [JH95]). The Evolutionary Line algorithm optimizes a set of initial, random candidates. The details of the algorithm like the encoding of the candidates, selection method, genetic operators (i.e., mutation, crossover), and a termination criterion will be described in Section 3.

Figure 1 illustrates a comparison between our evolutionary algorithm and an *exhaustive exploration* of Ω starting with a dense regular streamline sampling followed by filtering. Here, our approach performs less integrations and achieves a better visual result than the brute-force method.

3. Algorithmic Details

NOTATION The following notation is used throughout the paper. We refer to a single solution candidate as *individual* I. The fitness of a specific individual I_i is referenced as $f(I_i)$. Furthermore, n_p of these individuals form the *population* P. For steering the algorithm's behavior, percentages for *Elite Selection* p_e, mutation p_m, *Crossover* p_c, and *Insertion* p_i are used with respect to n_p.

INITIALIZATION During the initialization the population P of size n_p is created. Each individual $I \in P$ represents one solution candidate encoded by its genome the seeding position $\mathbf{p} \in \mathbb{R}^3$. The initial population contains only individuals with random genomes.

EVALUATION During this step the fitness function is evaluated and assigned to all individuals. With this, we obtain a value $f(I_i)$ with $i \in [0 \ldots n_p - 1]$. Therefore, we use two different functions, namely line arc-length and line curvature evaluated on the streamline integrated numerically for each individual.

FITNESS-BASED SORTING After the evaluation the list of individuals is sorted according to their fitness. Subsequent algorithm steps, such as elite selection, mutation, crossover, and the calculation of convergence measures rely on an ordered list of individuals.

ELITE SELECTION Keeping the n_e best individuals for the next generation P' is referred to as elite selection. This ensures that an already good solution is kept between generations. Individuals are selected from the ordered list according to I_i with $i \in [0 \ldots n_e - 1]$ and $n_e = p_e \cdot n_p$.

CROSSOVER The first genetic operator combines the genome of n individuals to create n offsprings. In its simplest form $n = 2$, each offspring inherits half of each parents' genetic information. In current implementation we do not use the crossover operator since our genome only contains three coordinates and therefore it would lead to inserting copies of individuals in the population.

MUTATION The second genetic operator alters each individuals genome slightly, whereas the probability, the type and strength of mutation is configurable. In our implementation we add a weighted displacement vector $\mathbf{d} \in [-1.0, 1.0]^d$ to the individuals genome; here d is the dimension of the search space. Futhermore, we facilitate a generation dependent weight which decreases with increasing generation count by a constant factor. This ensures that with increasing average fitness of the population good solution candidates are less altered. Individuals are selected from the ordered list according to I_i with $i \in [n_c \ldots n_e - 1]$ and $n_c = p_c \cdot n_p$, $n_e = p_e \cdot n_p$ The parameters are all kept constant throughout the paper.

INSERTION During this step a part of the population is replaced. Therefore, $p_i \cdot n_p$ individuals in P are replaced by new individuals when inserted in P'. Similar to the initialization the new individuals genome is randomly generated. This genetic diversity ensures further random exploration of the search space.

4. Experiments

For the demonstration of the effectiveness of the proposed method two data-sets related to technical and medical flow simulation have been chosen. At first a simulation of the well-known Kármán Vortex Street is used [CSBI05]. Secondly, an aneurysm simulation from the CFD Rupture Challenge 2013 [BRB*15] is used. Besides the different target functions and the constrains the same settings where used for both data-sets and are listed in Section 5. Throughout this paper the algorithm was configured with $n_p = 100$, $p_e = 10\%$, $p_c = 0\%$, $p_m = 45\%$, $p_i = 45\%$

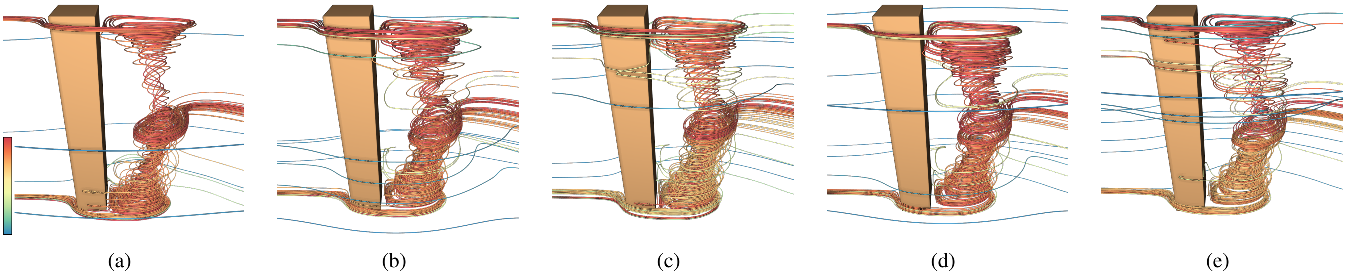


Figure 2: *Five different solutions of our method after 15 iterations for a timestep of the Kármán Vortex Street. Settings and initial conditions for the algorithm are identical. Images show the area behind the cylinder. Color encodes line arc-length.*

KÁRMÁN VORTEX STREET: The data-set represent a simulation of a three-dimensional flow around an obstacle, in our case a squared cylinder. The left boundary plane represents the inflow and the right boundary plane the outflow area. The search space Ω is defined by the two-dimensional inflow plane and the fitness function is the geometric length of the streamline, which is integrated until it leaves the domain. This choice is based on the assumption, that a streamline is longer if passing a vortex.

Figure 1 compares the evolutionary lines with a standard filtering approach. The result of the new algorithm clearly outperforms the standard method while computing less than half as much streamlines. The flow behavior close to the vortex is better expressed while still giving some context of the surrounding flow. Since the optimization is a guided random process it not always converges to the same result however with a constant quality. Figure 2 illustrates the variation of the results for identical initial conditions, fitness function and algorithm configuration.

FLOW IN AN ANEURYSM: CFD CHALLENGE CASE 1: The Simulation depicts the blood flow within an aneurysm data-set presented for the CFD Challenge in 2013. Here, complex flow structures are formed within the aneurysm sack. The search space Ω is here the entire three-dimensional domain. Streamlines are integrated in backward and forward direction until they leave the domain. An additional termination criterion is a maximum number of integration steps to avoid numerical issues close to zeros in the data. Due to the irregular domain streamline length is not an optimal choice as fitness function. Instead the average line curvature is a good measure to find streamlines passing vortices and has been used as fitness function.

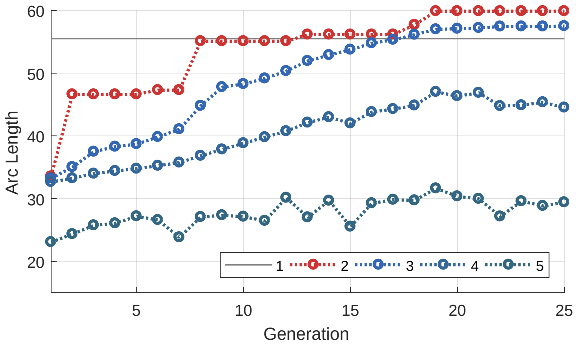


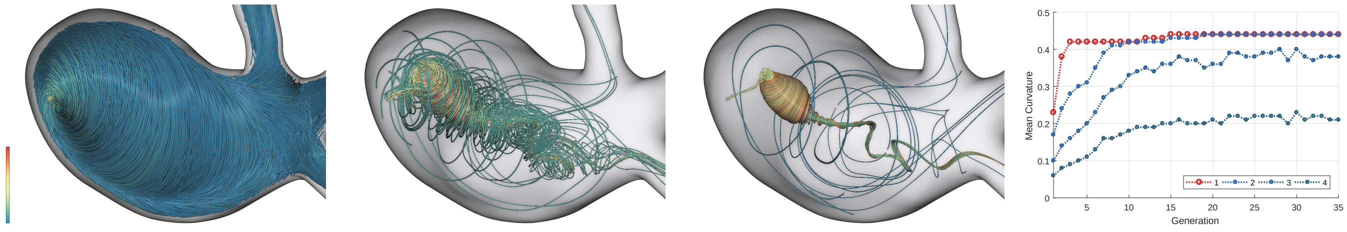
Figure 3: *Convergence behavior of the optimization process for the Kármán Vortex Street. (1) shows the maximum fitness for a dense seeding with 5000 streamlines. (2) maximum fitness of population. The average fitness for 10 (3), 50 (4) and 100 (5) percent of the ordered population are shown in blue colors.*

Figure 4 compares the evolutionary lines with a filtering approach using line curvature. The dense line set in Figure 4(a) consists of 5000 streamlines of which 97 (1.94%) are kept after filtering as shown in Figure 4(b). In contrast, Figure 4(c) shows evolutionary lines using a population of 100 individuals using 20 iterations. With this, we were able to clearly extract the core of the complex vortex near the center of the aneurysm sack.

5. Analysis

CONVERGENCE: The analysis of the convergence behavior gives insight into the mode of operation of the evolutionary process. Figure 3 shows the populations' fitness value for the Kármán Vortex Street in dependence of the number of iterations. Plotted are the maximum fitness (2) in red and three averages over 10%, 50%, and 100% of the population size in blue (3,4,5). The maximum fitness value achieved using a random dense seeding of 5120 lines has been plotted as reference in grey (1). The evolutionary algorithm tops this value already after 12 iterations. The 10% best lines are converging rapidly against this value. The average fitness of all lines does never exceed more than half of the maximum value due to the 45 percent newly generated lines in each iteration. Those lines can be used as context to the feature lines with the highest fitness value. A similar behavior but even faster convergence can be seen for the aneurysm data, Figure 4(d). Here, one can observe the fast increase in overall fitness already during the first eight iterations. After iteration 18 the fitness of the best 10% is identical to the maximum fitness.

POPULATION SIZE: The population size is one of the most essential parameters of the algorithm. To understand the influence of this parameter we have made a study measuring the number of required iterations to reach a benchmark streamline quality. Therefore, we let the algorithm run until a specific number of streamlines with fitness value higher than a given value is reached. Thereby, we consider the standard filtering approach as a special case with a



(a) (b) (c) (d)

Figure 4: *CFD Challenge Case 1: (a) Exhaustive exploration of the search space by seeding 5000 streamlines within the domain. (b) Resulting set after filtering contains 97 lines. (c) direct optimization of 100 individuals after 20 iterations. (d) Convergence of optimization process with maximum fitness (1), average fitness of 10 (2), 50 (3) and 100 (4) percent of the ordered population for 35 iterations.*

high population size applying one iteration. As a number of streamlines we have chosen 14 and the fitness value is set to 80% of the maximal value in the data-set. The results are plotted in Figure 5. This plot shows that in the optimal case for a population of 100 an efficiency gain of one order of magnitude can be achieved. The corresponding images can be found in the supplementary material.

PERFORMANCE: The performance of the evolutionary lines is dominated by the evaluation of the fitness function, which is the streamline integration and predicate computation. Thus, a fair comparison with standard filtering methods, which also require streamline integrations and predicate computations, is the total number of integrations necessary to reach comparable results, Figure 5.

6. Comparison

META-HEURISTICS: Simulated annealing (SA), evolutionary algorithms (EA), or hill climbing are referred to as meta-heuristics used if the target function is discrete or the search space very large. These algorithms aim for finding a close to optimal solution. Hill climbing as a deterministic process can get stuck in local extrema. Where in contrast SA and EA accept temporarily worse candidate during the optimization process. The general difference is the usage of a single candidate solution (SA), or a population (EA) which allows for a faster exploration of the search space, higher flexibility provided by genetic operators and it is parallelizable.

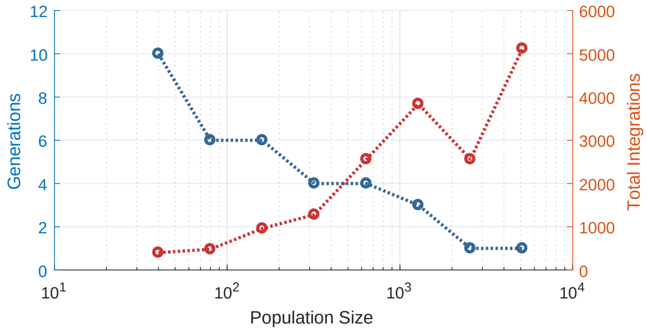


Figure 5: *Correlation of population size and required streamline integrations for Kármán Vortex Street. In the optimal case of a populaton size of 100 the efficiency gain is one order of magnitude.*

STREAMLINE PLACEMENT: Most streamline based methods use either intelligent placement or selection to find expressive sets. Typical approaches for selection are either based on similarity measures [MJL*13] or information theory [XLS10]. Such methods have the advantage that they allow for large varieties of visualizations, including viewpoint dependency [TMWS13] but they require the pre-computation of large line pools. Placement methods generally follow different goals and consider mostly seeding points only. Our method avoids the expensive pre-computation, it can be directly applied to an unknown data-set and can be parallelized since the solution candidates do not exchange information.

7. Discussion and Conclusion

With this paper we presented first experiments with an evolutionary algorithm in the context of geometry-based flow visualization. We tested our approach for streamlines in steady vector fields and carefully analyzed convergence and parameters. While the chosen examples are simple, they clearly demonstrate the potential of the approach with efficiency gains up to one order of magnitude. So far, we did not yet exploit the full power of the framework of evolutionary algorithms, e.g the mechanism of *CrossOver* which become interesting for complex genomes. For increasing complexity of the fitness function and genome it is to expect that even higher efficiency improvements or even solutions that are unfeasible with classical approaches are achieved. As such we plan to investigate time-dependent data-sets i.e., optimize sets of path, time, and streak lines with a four dimensional search space and evaluate useful flow and line measures which can serve as fitness function for this. A further interesting field is the optimization of surface structures.

Besides the promising results discussed above we have also observed similar cases which cannot directly be solved within our framework. The algorithm, in its current form, is targeted towards finding isolated maxima. As soon as one candidate falls in the neighborhood of a dominant maximum the candidates from other less expressed maxima are running the risk of extinction. This means handling ridges in the fitness function landscape is not straight forward. We consider it as a challenging task for future investigations to develop evolutionary methods for the extraction of extremal structures, e.g., in derived scalar fields like FTLE or λ_2. Finally we will study possibilities for a GPU-based implementation of our algorithm for handling large population sizes as well as improve run-time of the optimization process.

References

[Bäc96] BÄCK T.: *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press, 1996. 1

[BRB*15] BERG P., ROLOFF C., BEUING O., VOSS S., SUGIYAMA S.-I., ARISTOKLEOUS N., ANAYIOTOS A. S., ASHTON N., REVELL A., BRESSLOFF N. W., ET AL.: The computational fluid dynamics rupture challenge 2013 phase II: Variability of hemodynamic simulations in two intracranial aneurysms. *Journal of biomechanical engineering 137*, 12 (2015), 121008. 2

[BS93] BÄCK T., SCHWEFEL H.-P.: An overview of evolutionary algorithms for parameter optimization. *Evolutionary computation 1*, 1 (1993), 1–23. 1

[CSBI05] CAMARRI S., SALVETTI M.-V., BUFFONI M., IOLLO A.: Simulation of the three-dimensional flow around a square cylinder between parallel walls at moderate Reynolds numbers. In *XVII Congresso di Meccanica Teorica ed Applicata* (2005). 2

[Dar59] DARWIN C.: *The Origin of Species by means of Natural Selection, or, the Preservation of Favoured Races in the struggle for life*. John Murray, 1859. 1

[GRT13] GÜNTHER T., RÖSSL C., THEISEL H.: Opacity optimization for 3d line fields. *ACM Trans. Graph. 32*, 4 (2013), 120:1–120:8. 1

[GRT14] GÜNTHER T., RÖSSL C., THEISEL H.: Hierarchical opacity optimization for sets of 3d line fields. *Computer Graphics Forum 33*, 2 (2014), 507–516. 1

[JH95] JEONG J., HUSSAIN F.: On the identification of a vortex. *Journal of Fluid Mechanics 285* (2 1995), 69–94. 2

[KBB*16] KRUSE R., BORGELT C., BRAUNE C., MOSTAGHIM S., STEINBRECHER M.: *Computational Intelligence: A Methodological Introduction, 2nd Edition*. Springer, New York, 2016. 1

[LMSC11] LEE T.-Y., MISHCHENKO O., SHEN H.-W., CRAWFIS R.: View point evaluation and streamline filtering for flow visualization. In *Proc. of IEEE PacificVIS* (2011), pp. 83–90. 1

[MJL*13] MCLOUGHLIN T., JONES M., LARAMEE R., MALKI R., MASTERS I., HANSEN C.: Similarity measures for enhancing interactive streamline seeding. *IEEE Transactions on Visualization and Computer Graphics 19*, 8 (2013), 1342–1353. 4

[MLP*10] MCLOUGHLIN T., LARAMEE R. S., PEIKERT R., CHEN M., POST F. H., CHEN M.: Over two decades of integration-based, geometric flow visualization. In *Computer Graphics Forum - State of the Art Reports* (2010), vol. 29, pp. 1807–1829. 1

[SGSM08] SALZBRUNN T., GARTH C., SCHEUERMANN G., MEYER J.: Pathline predicates and unsteady flow structures. *The Visual Computer 24*, 12 (2008), 1039–1051. 1

[TMWS13] TAO J., MA J., WANG C., SHENE C.: A unified approach to streamline selection and viewpoint selection for 3d flow visualization. *IEEE Transactions on Visualization and Computer Graphics 19*, 3 (2013), 393–406. 4

[XLS10] XU L., LEE T.-Y., SHEN H.-W.: An information-theoretic framework for flow visualization. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (2010), 1216–1224. 4