Genetic Algorithm Approach to Path Planning for Intelligent Camera Control for Scientific Visualization

Djamalladine Mahamat Pierre and Nordin Zakaria

Universiti Teknologi PETRONAS,
Bandar Seri Iskandar, 31750 Tronoh, Perak, Malaysia
nordinzakaria@petronas.com.my,
djamal2810@gmail.com
http://www.utp.edu.my

Abstract. In this paper, we propose to develop an intelligent camera control algorithm for scientific visualization. Intelligent camera control refers to a path planning algorithm that allows a virtual camera to navigate a scene autonomously. Intelligent camera overcomes some shortcomings of traditional manual navigation such as the risk of getting lost in the scene, or the user's distraction from the main goal of the study. In the past years, several path planning approaches have been proposed. While those approaches focus on determining the shortest path between two points, they cannot adapt to multiple constraints that a virtual camera is subjected to, in scientific visualization. Inspired by Unmanned Aerial Vehicle path planning, our algorithm uses genetic algorithm as an optimization tool. Finally, the paper presents the experimental results of our algorithm including an empirical study to determine the optimal values for the genetic parameters.

Keywords: Intelligent Camera Control, Genetic Algorithm, Path Planning.

1 Introduction

Scientific visualization is used in many areas ranging from chemistry to medicine, astronomy, fluid mechanics, and volume rendering. Many researches resulted in various approaches to improve scientific visualization. For intense, the term "intelligent camera control [8]" is introduced by Steven M. Decker and refers to a virtual camera capable of exploring a virtual environment autonomously. The goal of the intelligent camera is to determine a collision free short pathway between the camera's current position and a destination within an acceptable time frame. In past years, various optimization approaches to path planning have been used to determine the shortest path between two points. The approaches include potential field, cubical path, Randomized Rapidly-Exploring Random Tree, neural network, and recently, genetic algorithm.

Steven M. Decker designed a framework for path finding using potential field [8]. While this approach does not clearly handle local minima, Decker, in his subsequent works in [9], proposed several ways to handle it. Although this approach efficiently provides the shortest path between two points, it is not robust and flexible to adapt to multiple constraints that the virtual camera can be subjected to, in visualization.

Potential field assigns attraction values to the objective (the destination point) while repulsive values are assigned to the obstacles. The camera is pulled by the attraction force at the objective, while it is pushed away from the obstacles by the repulsive forces. The CubicalPath, presented by Beckhaus et al. at [6] and [10] discretizes the scene into a cube space (coarse voxel) where attraction values assigned to the cubes dictate the motion of the camera. Both potential field and cubical path suffer from unhandled local minima and forces compensation problems. When those states occur, the camera is virtually stopped from reaching its goal.

Graham et. Al [4] exploited path finding algorithm using neural network. Leigh et. Al [5] describes an alternative to A*-Like path finding using genetic algorithm. Their algorithm appears more efficient with multiple agents. Unfortunately, it relies on 2D testing ground. Expending the use of the algorithm to 3D environment does not necessary produce the same success level.

Hugen et al.[11] provides a path planning approach for mobile robots using genetic algorithm. Sensors are used to locate the obstacles while the obstacle response is computed locally. The algorithm can fit in virtual environment with multiple moving obstacles. The only difference is the use of ray casting in virtual environment instead of sensors in real world. However, this algorithm has some shortcomings. While the robots operate in a 3D real world space, the path planning is computed for a 2D world assuming that the robots operate on a flat floor. Moreover, although the algorithm can operate in a completely unknown environment, it is not concerned with the visibility of the objective.

[1, 2, and 3] describe a path planning approach for Unmanned Aerial Vehicle (UAV). The approach relies on genetic algorithm for path optimization. Genetic Algorithm (HOLLAND, 1975) is a search heuristics that imitates evolution in nature. Although the algorithm presented in [1, 2, and 3] does not handle the early visibility of the objective, the virtual camera and the UAV have some similarities on their behaviours: finding a collision free pathway in a three-dimensional environment.

While most of the available paths planning algorithms focus on determining the shortest path, their applicability in scientific visualization is narrow. We are proposing an algorithm that takes into consideration, not only the length of the path, but also an early visibility of the objective of the camera, into consideration. In the following lines, objective refers to a point of interest in the virtual environment, which the camera should focus on at destination. The goal of our intelligent camera is to reach and set focus on the objective as early as possible.

2 Problem Statement

Manual navigation (using mouse, keyboard, joystick) is the most commonly used mode of navigation for virtual scene exploration. Although manual navigation gives user the freedom to explore the scene and learn, it has a lot of shortcomings:

- In a relatively complex scene, exploring the scene becomes difficult. Users may find it difficult to locate particular point of the scene (containing multiple obstacles).
- Manual navigation may distract users and take their attention away from the point of study.

Besides manual navigation, autonomous agents are also used to improve navigation within a virtual scene (intelligent camera control). However, those agents are more concerned about finding the shortest path rather than putting a particular interest on some areas of the virtual scene: which is one characteristic of scientific visualization. Figure 1 depicts two possible path ways for an intelligent camera. The upper path way is longer and allows the camera to view the objective at a time T_1. The lower pathway is shorter but allows the camera to view the objective only at time T_2 greater than T_1. This example shows that finding the shortest path for a virtual camera is not enough to fulfill the requirements for the scientific visualization.

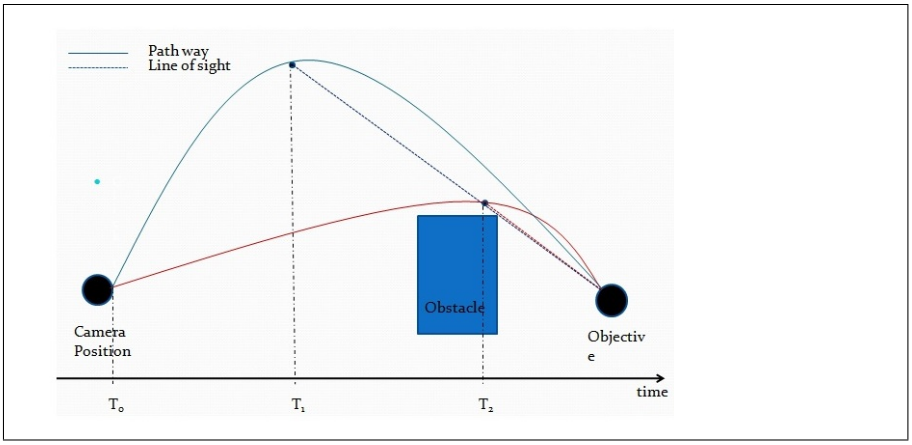


Fig. 1. Comparison of optimal path against shortest path

3 Objective

The objective of this research is to develop an algorithm capable of:

- Finding a collision free pathway from a virtual camera's current position to a destination.
- Ensure the pathway has an acceptable length and should be computed within an acceptable time frame.
- Keeping track of the destination point by ensuring that it is kept on the camera's field of view throughout the simulation.

4 Design and Implementation

4.1 Path planning

As a 'flying agent', our intelligent camera shares some similarities with a UAV. Therefore, the genetic algorithm based approach to path planning fits our algorithm better. The algorithm takes as parameters the camera's characteristics (position, orientation), the position of the objective, and the geometry of the scene (models and obstacles). The obstacles are created using triangle meshes. The path way is a B-Spline curve. The coordinates of the control points of the curve are passed to the GA as genes of a chromosome.

4.2 B-Spline

The phenotype of each individual is a B-Spline curve. A B-Spline curve is defined by a set of a few control points. The modification of the control points influence the shape of the curve. Those characteristics of the curve justify its choice over other curves'definitions which require more points, and therefore, require more memory space for computation. B-Spline curve is a parametric curve. The position of a point of the curve at a time t, in 3D space, is given by the following parametric function described [7]:

$$X(t) = \sum_{i=0}^{n} x_i B_{i,k}(t),\ Y(t) = \sum_{i=0}^{n} y_i B_{i,k}(t),\ Z(t) = \sum_{i=0}^{n} z_i B_{i,k}(t), \qquad (1)$$

where $B_{i,k}$ is the blending function of the curve and k is the order. K represents the smoothness of the curve. A higher value of k provides a smoother curve. The curve representing the path of the camera is computed using a finite number, n+1, of control points . The value of t ranges from 0 to n-k-2 and is incremented at constant step. The definition of $B_{i,k}(t)$ is done recursively in term of knots value. Each knot is represented by the following function described in [1, 2, 9]:

$$Knot(i) = \begin{cases} 0, & \text{if } i < K \\ i - K + 1 & \text{if } K \leq i \leq n \\ n - K + 2 & \text{if } n < i \end{cases}$$

The definition of Bi, k (t) in term of the knot values is given by [1, 2, 9] as:

$$B_{i,1}(t) = \begin{cases} 1, & \text{if } Knot(i) \le t < Knot(i+1) \\ 1, & \text{if } \begin{cases} Knot(i) \le t \le Knot(i+1) \\ \text{and} \\ t = n - K + 2 \end{cases} \\ 0, & \text{otherwise} \end{cases}$$

$$B_{i,k}(t) = \frac{(t - Knot(i)) \times B_{i,K-1}(t)}{(Knot(i+K-1) - Knot(i)} + \frac{(Knot(i+K) - t) \times B_{i+1,k-1}}{Knot(i+K) - Knot(i+1)}, \quad (2)$$

4.3 Genetic Algorithm

Genetic algorithm is a search heuristic that imitates evolution in nature. A potential solution, or individual, is represented by a chromosome. Each chromosome may contain several genes representing the characteristic of the individual. Starting from a couple of potential solutions, the individuals go through some transformations to generate a new and fitter individual. The transformations include reproduction, crossover and mutation. Reproduction involves making a copy of a chromosome. Crossover changes the content of individual by swapping the values of genes between two chromosomes. This approach mimics 'mating' of the individuals involved [12]. Mutation, on the other hand, alters the value of a gene to generate a new individual. If an individual is judged unfit in the process, it is simply discarded. The judgement on fitness of the individuals is based on a value (fitness value) computed using a fitness function. The fitness function is defined based on characteristics of the chromosome, or genes' values.

While obstacles in [1, 2, and 3] are represented by a specific function, in our testing ground, they are represented by triangle meshes. Ray casting is used to determine the position of the obstacles and test the validity of the control points. The coordinates of the control points represent the genes of chromosomes or individuals. Chromosomes are evaluated using a fitness function f. The fitness function is inversely dependant on a sum of terms f_i. Each term f_i represents a penalty, or the extent to which the virtual camera is far from meeting a particular constraint. The following formula is a representation of the fitness f:

$$f = 1/\sum_{i=1}^{c} a_i f_i \quad (3)$$

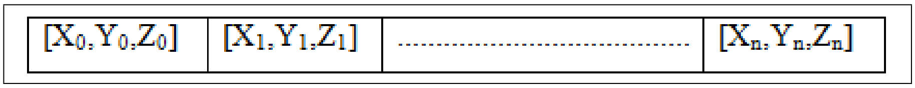
where c is the number of constraints, a_i is the weight of the term f_i. An individual with lower f_i values has higher fitness values; therefore, a camera following a pathway derived from such individual is closer to meet the constraints.

Two fixed points represent the camera's initial position and the objective respectively. Because of the static nature of the geometry of the scene and the objective, it is possible to determine one optimal path for the camera using a fixed number of control points. When the objective is static, four constraints apply to the optimization problem.

- f_1 penalizes all segments of the curve that cross the obstacles. The fitter curves are those with less penalties. f_1 is the most important constraint because it validates all the feasible points of the path. The penalty is computed using two successive points, P(t) and P(t+1) determined by (1). A ray, originated at P(t) is casted towards P(t+1). If there is no feedback, than there is no obstacle insight. If there is a feedback and the distance between the P(t) and the obstacle is greater than the distance between P(t) and P(t+1), no penalty is given. That is because the segment delimited by P(t) and P(t+1) is not crossing the triangle hit by the ray. On the other hand, if the distance is smaller, a penalty is given by incrementing the term f_1. The smaller distance indicates that the triangle of the obstacle hit by the ray is located between P(t) and P(t+1); in other words, the segment is crossing the obstacle.
- f_2 penalizes all points from which the objective is not visible. In other words, f_2 help getting a clear line of sight with the objective and strives to kept it. f_2 determines the early visibility of the objective.
- f_3 is a special penalty related to the length of the path way. A bigger value of f_3 reflects a longer path from the camera's position to the destination.
- f_4 ensures that a premature convergence of the control points does not occur. In order to avoid the accumulation of the control points in a location, a safe distance is maintained between two consecutive control points. f_4 indirectly penalizes local optima.

4.3.1 Genes

In our problem, the coordinates of the control points represent the genes. Since all the control points should be taken into consideration for the curve definition, the expected solution is a set of the same number of control points. If we have n+1 control points, then they are represented by genes indexed by 0 through n as depicted in the Figure 2.



[X_0, Y_0, Z_0]	[X_1, Y_1, Z_1]	…………………………………	[X_n, Y_n, Z_n]

Fig. 2. Representation of a chromosome with a series of coordinates as its genes

X_i, Y_i, and Z_i, in Figure 2, represent the coordinates of the control points in 3D space coordinates.

5 Results

Our testing ground is a virtual environment with models. The models, representing the obstacles, are made of triangle meshes. Figure 3 shows a the virtual environment with buildings. The starting point and destination of the virtual camera are on either sides of the group of buildings.

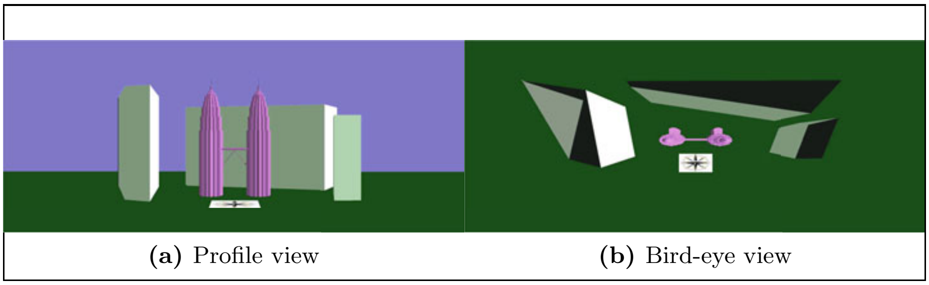


Fig. 3. A two-perspective view of the virtual environment

With the default value of f_1 kept to 1, the maximum possible fitness that can be (logically) obtained is 1 (1/(1)). That is (logically) possible if and only if the penalties from all the terms f_2, f_3 and f_4 are null; in other words, there is no obstacle between the start point and the destination of the camera. A simple test is conducted to observe the phenotypic change of the pathway. With a crossover rate of 50 percent, a mutation rate of 20 percent, and the population size of ten (10), the fitness value of the fittest individual,after one hundred (100) generations, is 0.3264.

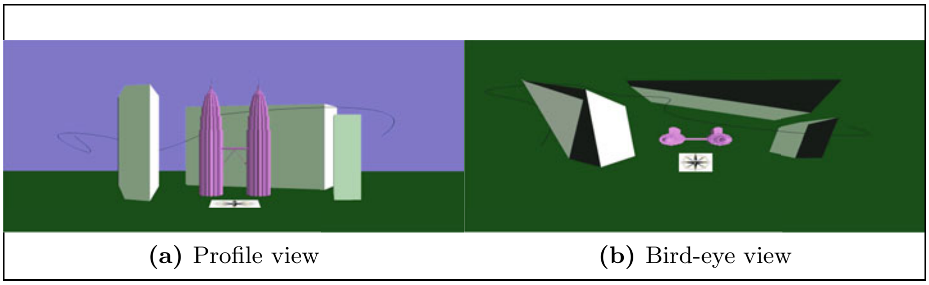


Fig. 4. Profile and Bird-eye views of a pathway with a fitness value of 0.3264

The relatively low fitness value can be justified by the length of the generated pathway as well as the obstraction of the objective from each discrete point of the curve. We can observe that the pathway links the start point to the destination without touching or crossing the models (buildings). However, The pathway sometimes presents unessary curves, which increases its lengths, and consequently, decreases its fitness. On the otherhand, from most discrete points of the pathway, it is impossible for the camera to perceive the objective, which also has a negative effect on the fitness.

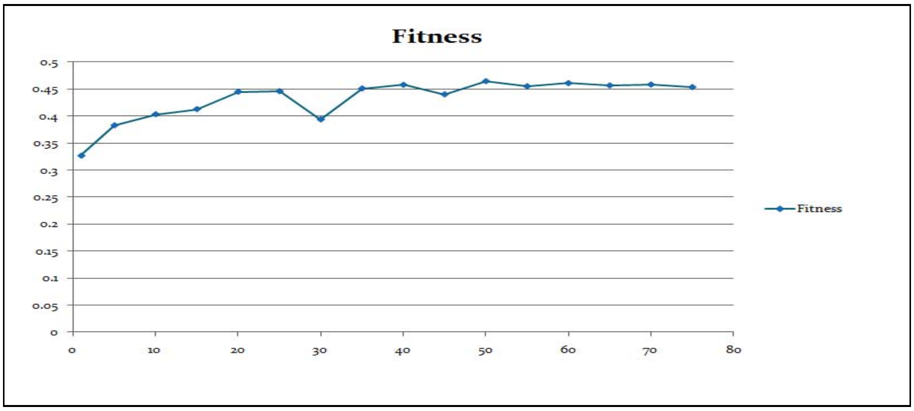


Fig. 5. Variation of fitness value with respect to the population size

6 Conclusion and Observation

There has been a lot of debate on what is the appropriate values for the crossover or mutation rates to have an optimal individual. Some experts suggest that, because those values may vary according to the problem in hand, tuning can be used to determine them. However, empirical studies conducted in our test case are inconclusive. First, the probabilistic characteristic of genetic algorithm and the random generation of genes during mutation or the creation of the population makes it hard to anticipate the impact of those values for the next test case. Dumitrescu et. al [13] states that a higher propulation size provides larger variety, and consequently, a bigger chance of exploiting the different individuals to determine a more reliable result. That statement is verified only when the individuals of the initial population are scattered and occupy a large space within the search space. But, if the initial population is generated randomly, there is guarantee that the individuals will be scattered within the search space. A series of tests, conducted to depict a relationship between the population size and the fitness values, is shown at Figure 2.

A general view of the graph shows that the fitness value increases as the size of the population rises. However, the sudden drops of the fitness at 30, 45 and 55 proves the assumption wrong. This issue might be solved if the population is systematically scattered in the search space. In order to maintain consistancy, it is imperative to consider modifying the selection procedure. In a traditional genetic algorithm, the individuals that are subjected to genetic transformations are selected randomly. This may give the perception that the final result is not predictable. The main question remains if the radom characteristics of the selection procedure and the population creation is necessary. In a typical video games, it is important that the agents' behaviors are unpredictable. However, the same cannot be said about path planning as users would not mind if the camera uses the same pathway at different test cases as long as the fitness of the pathway is acceptable.

References

1. Nikolos, I.K., Tsourveloudis, N.C., Valavanis, K.P.: Evolutionary Alogrithm Based O_-Line Path Planner for UAV. AUTOMATIKA 42(3-4), 143–150 (2001)
2. Nikolos, I.K., Valavanis, K.P., Member, S., Tsourveloudis, N.C., Kostaras, A.N.: Evolutionary Alogrithm Based Of- ine/Online Path Planner for UAV Navigation. IEEE Transsactions on Systems, Man, nad Cybernetics-Part B. 33(6) (2003); Euro-Par (2006)
3. Nikolos, I.K., Tsourveloud, N.C., Valanis, K.P.: EvolutionaryAlgorithm Based 3-D Path Planner for UAV Navigation
4. Graham, R., McCabe, H., Sheridan, S.: Neural Networks for Real-time Path finding in Computer Games. School of Informatics and Engineering, Institute of Technology at lanchardstwon, Dublin 15
5. Leigh, R., Louis, S.J., Miles, C.: Using a Genetic Algorithm to Ex- plore A*-like Pathfinding Algorithms. In: Proceedings of the 2007 IEEE Symposium on Computational Intelligence and Games, CIG 2007 (2007)
6. Beckhaus, S., Ritter, F., Strothotte, T.: CubicalPath - Dynamic Potential Fields for Guided Exploration in Virtual Environments
7. Mittal, S., Deb, K.: Three-Dimensional Offline Path Planning for UAVs Using Multiobjective Evolutionary Algorithms. In: IEEE Congress on Evolutionary Computation, CEC 2007 (2007)
8. Drucker, S.M.: Intelligent Camera Control for Graphical Environments. In: partial fulfillment of the requirements for the degree of Doctor of Philosophy at Massachusetts Institue of Technology (June 1994)
9. Drucker, S.M., Zeltzer, D.: Intelligent Camera Control in a Virtual Environment
10. Beckhaus, S.: Dynamic Potential Fields for Guided Exploration in Virtual Environments. Dissertation (2002)
11. Burchardt, H., Salomon, R.: Implementation of Path Planning using Genetic Algorithms on Mobile Robots
12. Rotsan, N., Meffert, K.: JGAP Frequently asked questions. Copyright (2002- 2007)
13. Dumitrescu, D., Lazzerini, B., Jain, L.C., Dumitrescu, A.: Evolution Computation, 21–37 (2000)