Interactive Evolution in the framework of Simulated Natural Evolution

Jeanine Graf and Wolfgang Banzhaf

Abstract

This paper demonstrates how interactive evolution can be applied to the evolution of simulated natural evolution. Evolutionary algorithms of selection and variation by recombination and/or mutation can be used to simulate biological evolution in a computer. Interactive evolution can be used to direct the development of graphic models that are simulations of their counterparts in nature. Interactive evolution thus can be used to evolve models of animals and plants. We show that interactivity of artificial evolution can serve as a useful tool for achieving progress in the ontogenesis of simulated models and help paleontologist to solve problems in identifying likely missing links.

Keywords: Growth, Paleontology, Evolutionary Algorithms, Simulation of Natural Evolution, Computer Graphics.

1 Motivation

World view at the present time is influenced by the knowledge that the universe, the earth and all living things have evolved through a long history of continual, gradual change shaped by more or less directional natural processes consistent with the laws of physics (E. Mayr 1978).
Paleontologists would like to discover morphologic intermediates in the fossil record. There are still gaps between each of the groups of animals and plants, which motivated us to consider the problem of finding out some morphologic intermediates with the help of simulated evolution.

The novel idea in this paper is to provide the paleontologist with a new technique to evolve simulated models of animals and plants and try to visualize the evolution. We have developed a system that presents progressively evolving models for artificial life by interactive processes. The application of interactive evolutionary algorithms in computer graphics helps us to demonstrate continuous evolutionary change with.
Scientists or workers in the field of biology have some difficulties to conclude how the growth and expansion process evolved due to problems in observing the change in nature. We show that interactive evolution can serve as a useful tool for achieving flexibility and complexity in simulated evolution with some amount the scientific-feedback and detailed knowledge.

2 Evolution in Nature

Natural selection acts nearly without fail on individuals or on groups of related individuals. It is only during the last fifth of the history of life on the earth that multicellular organisms have existed. They seem to have taken shape from unicellular organisms on numerous occasions (J. Valentine, 1978)
The animals and plants are all multicellular, made up of millions or billions of individual cells. Even the simplest multicellular organisms include diverse categories and types of cells. At least two million multicellular species exist today, and many others have come and gone over the ages. The important advantages of multicellularity stem from the repitition of cellular machinery it implies. From this property or attribute derives the ability to live longer, to produce more offspring, to be larger and so to have a greater internal physiological stability, and to construct bodies with a variety of architectures.
Just as change in the environment can involve organisms, so can the activities of organisms involve the environment to create new conditions.
The history of life shows evidence of interaction between environmental change on the one hand and the evolutionary potential of organisms on the other. It is therefore of interest to briefly examine and keep in mind the major causes of environmental changes of biological relevance. Therefore we try to observe the evolution of different kinds of animals, and try to find out the continuity of their evolution.

3 Artificial Evolution

Natural evolution is a mechanism that can be simulated and endeavored. It is described by the term i.e., artificial evolution.
In nature many shapes change spectactularly in form. Evolution, however, implies change by continuity. Therefore complex processes of evolution can be sufficiently represented by different continuous transformations. D'Arcy W. Thompson recognized that the concept of geometric transformation could be helpful for identifying morphological change.
Some researcher and workers in the field of Biology try to find the ancestor and interconnection between species. We can observe that there is a line of continuity in the morphology of some animals. In that case, we try to observe these changes and the transition from one animal to another or from one species to another. It is therefore of interest to apply this knowledge as background for simulated evolution.
We are not able to say whether two or more animals, which look morphologically similar are descendants from the same ancestor. The evidence of convergent evolution is demonstrated by the indisputable fact that unrelated groups of animals do respond to similar selective pressures with similar adaptations. But let us try with help of interactive evolution, to simulate some biological evolution on the computer. We shall observe that individuals of related groups resemble each other.
Much of evolutionary biology is the working out of an adaptative program. Evolutionary biologists assume that each aspect of an organism's morphology, physiology and behavior has been modeled by natural selection as a solution to a problem posed by the environment (R. Lewontin 1978).
From these facts and from the wealth of accumulated evidence on the comparative embryology and morphology of the living representatives of fossil groups it is possible to build a picture of the rise of major animal groups. It is difficult to know the descendants of these animals, hence we use interactive simulated evolution to examine such problems.

4 Evolutionary Algorithms

Interactive evolution is a technique from the class of evolutionary algorithms (EAs), which are based upon a simple model of organic evolution. Most of these algorithms operate on a population of individuals that represent search points in the space of the decision variables (either directly, or by using coding mappings). Evolution proceeds from generation to generation by exchanging genetic material between individuals (*recombination*), i.e., by trying out new combinations of partial solutions, and by random changes of individuals (*mutation*). New variations are subjected to *selection* based on an evaluation of features of the individuals according to certain (*fitness*) criteria.

The best-known representatives of this class of algorithms are evolution strategies (ESs), developed in Germany by I. Rechenberg (21) and H.–P. Schwefel (23), evolutionary programming (EP), developed in the U.S. by L.J. Fogel (Fogel, et al. 1966), and genetic algorithms (GAs), developed in the U.S. by J.H. Holland (17).

Evolution strategies (ES) were first applied in the field of experimental optimization and used discrete mutations (Rechenberg 1994). When computers became available, algorithms were devised that operated on continuous-valued variables. ES uses random mutation as its main search operator. But in addition, recombination operators are also included, at the discretion of the human operator. Parameters for self-adaption of the step sizes are also provided, although the methods for performing self-adapation (Schwefel 1995) precede those of Fogel et al. (1992), and initial comparisons favor the methods of Schwefel (1981). Evolution strategies are often described in the form (μ+λ), in which μ parents give rise to λ offspring and the best μ solutions from both parents and offspring become parents of the next generation, or (μ, λ), in which the best μ solutions are chosen solely from the λ offspring, and the old parents are replaced (see Rechenberg 1973; Schwefel 1981; Baeck 1994).

Evolutionary programming (EP) was originally proposed as a method to optimize finite state machines for time series prediction (Fogel et al. 1966), this as a prerequisite for evolving intelligent behavior. More recently, EP has been used to address real-valued optimization problems. A population of candidate solutions to the task at hand is subjected to random variation and selection over successive iterations. For real-valued problems, the variation usually takes the form of zero mean Gaussian perturbations, with the variances of the perturbations either being set as a function of the parent's error score (Fogel 1995), or as a self-adaptive value (Fogel et al. 1992). Recombination is not used in typical implementations of EP. Selection is accomplished probabilistically using a competition procedure that allows less-fit solutions some probability of being included in the subsequent generation, yet also guarantees that the best solution found to that point will be retained.

Genetic algorithms (GAs) emphasize the genotypic level of evolution, primarily operating on binary encodings (a choice that is founded by the argument of maximizing the number of schemata available; see Holland 1975; Goldberg 1989; Davis 1991). Selection is often made probabilistic in proportion to a solution's relative fitness; the greater the realtive fitness, the greater the probability of generating offspring. Emphasis is placed on using crossover operators to search for new solutions by bringing together pieces of different parents. The effectiveness of many of these crossover operators depends on the degree to which the components of a solution are independent. A low rate of mutation is applied to ensure that all solutions are possible in the limit.

In all of these cases, the selection criteria are traditionally fixed and are held constant from the start of the simulation, therefore these criteria must be detailed explicitly beforehand. This constitutes a significant problem in many realistic applications (apart from optimization), because an explicit fitness function may not be available in closed form. Recently, various work-arounds have been tried, one of the most prominent being *co-evolution*. In this method, rather than using one population to search for the best solution, two or more antagonistic populations are run which compete against each other. The realized fitness in this case is in part determined by the relationship of one population to the other, and does not have to be defined explicitly beforehand (16).

This paper makes use of an alternative method to generate fitness by incorporating the scientist into the selection process of artificial evolution. Our work in computer graphics, a natural domain for humans, easily engages the user by relying on human visual capacity.

5 Interactive Evolution

Dawkins (1986) demonstrated convincingly the potential of Darwinian variation and selection in graphics. He evolved *biomorphs* 2-D graphic objects, from a collection of genetic parameters with the user (7; 25). Recently, much research has been directed into the application of genetic algorithms to image and graphics problems. Sims (1991) used genetic algorithms for interactive generation of color art; Todd and Latham (1991) have considered similar ideas to reproduce computer sculptures through structural geometric techniques.

The idea offered in this paper is to provide a user with a new technique to evolve biological individuals models that can be applied in the field of biology. We have developed a system that presents progressively evolving solutions for paleontology and ontogony problems by means of interactive processes. In *interactive evolution*, the user selects one or more individual(s) which survive(s) and reproduce(s) (with variation) to constitute a new generation. These techniques can be applied to the production of computer graphics, animation, creating forms, textures, and motion . Potential applications of interactive evolution include artificial life design, e.g., development of components of biological nature and engineering construction design.

Phenotypes and Genotypes

We shall need to discern between genotypes and phenotypes in interactive evolution; both terms are also basic concepts for biological evolution. The biological genotype is the information that codes the development of an individual. Genotypes most often consist of DNA sequences. In this interactive evolution, genotypes are represented as numerical data and real values. The phenotype is the realised behavior of the individual itself, i.e., the product of an interpretation of the underlying genotypic representation. In our case, the phenotype is the resulting graphical model.

The relation between genotypes and phenotypes in nature is determined by the genotype-phenotype mapping. This transformation is very complicated and draws heavily from the individual's current environment. In principle, a similar mapping can be introduced in artificial evolution (3). In this way, a part of the complexity of the developing solution might be rendered by the environment.

Fitness and Selection

The term *fitness* in interactive evolution is the capability of an individual or model to survive into the next generation, and therefore is tied directly to selection. Usually, fitness is not defined explicitly but is instead a relative measure of success following from the selection activity of a human scientist. Here, it is even based on non-quantifiable measures like certain predefined criteria (for example: the historical knowledge about existing recombination between two defined animals) help to sort candidates for survival from the set of all variants, among which a human user ultimately determines the next generation.

Variation

In interactive evolution, one of the main benefits is the automatic generation of variants. Variation is accomplished by defining problem-specific mutation and recombination operators that constantly propose new variants to the presently existing population of simulated graphic models on the screen. An important amount of knowledge has to be invested in order to find appropriate operators for an application domain. The next section provides appropriate operators for manipulation of simulated animals or plants models.

5.1 Application of Interactive Evolution to the Simulation of Natural Evolution

The fossil record shows us many noticeable aspects, which we can use for interactive evolution. Those aspects which must be explained by any good model of the development of life on earth. Animal groups are appearing in a certain unmistakable order. For example, primitive fish appear first, amphibians later, then reptiles, then primitive mammals, then (for example) legged whales, then legless whales. In the next section we show an example for this transformation from legged to legless whale, see figure 6. This temporal-morphological reciprocality is very spectacular and impressive. It is important to apply this predicate order in interactive evolution. Many chains of groups contain one or more species which appear to link early, primitive genera with much more recent, radically different genera (e.g. reptile- mammal transition, hyenids, horses, elephants), through major morphological changes. Even for the spottiest disparities, there are a few separated intermediates that show how two apparently very different groups could, in fact, be related to each other (ex. -Archeopteryx-, linking reptiles to birds).

Paleontologists have observed transitions of at least 30 genera in nearly perfect morphological order, with most of the reptilian first and most of the mammalians last, and with only relatively slight morphological differences separating each successive genus.

There are still unknown species-species transitions, and the chain of genera is not complete, but we now have at least a partial lineage, and sure enough, the new whale fossils have legs, exactly as predicted. (for discussions see Berta, 1994; Gingerich et al. 1990; Thewissen et al. 1994; Discover magazine, Jan. 1995).

6 Evolving Simulated Models

In simulated evolution, one approach is to consider a growing object as a set of points. Each point can be defined using three coordinates, x, y and z. In artificial evolution, the process of altering or of growth can be systematically implemented using the concept of geometric transformations. A transformation is determined by defining new variables, x′, y′ and z′, as functions of x, y and z. A geometric transformation can often be considered as a general way of change, independent of the special object to which is then applied.

We describe methods for evolving plants and animal models based on distributive data interpolation. The methods use corresponding points in the images to be interpolated.

In interactive evolution techniques which can be used for object metamorphosis include solid deformations. In such a case, the 2-D or 3-D model of the first object is transformed in order to assume the shape and properties of the second model, and the resulting animation is recorded. The interactive evolution process helps us to construct body with varying architectures, which are potentially real shapes from nature.

Our approach evolves *two-dimensional* images specified either directly as bitmaps or as parameterized geometric models, such as those provided by vector graphics.

Bitmaps and other forms of direct encoding of images have found an excellent niche in computer graphics, video composition and image rendering (11; 18). Any 2-D shape can be represented as a sequence of points or vertices, with each vertex consisting of an ordered pair of numbers (x, y), its coordinates. The array of pixel values of a 2-D image, however, have nothing to do with the structure being represented in the image. This constitutes the challenge of finding appropriate operators for the generation of new variants of an existing image, because structural or functional conservation of the image content is of most importance in applications.

We solve the problem for realizing evolutionary operations by using *image interpolation* to create variations. Warping and morphing are methods which, by using tiepoints in two images "A" and "B", allows for the creation of intermediate images (30; 22). Basically, these intermediate images are interpolations along an abstract axis from image "A" and image "B".

In our interactive evolution, the tiepoints are the genotype and are constraints for the interpolation process, since corresponding tiepoints in "A" have to be transformed into those of "B". The method requires distributing tiepoints over two or more models in such a way as to conserve essential structures in interpolated (intermediate) models. In this way, an arbitrary initial model can evolve via intermediate steps of interpolation into a final model.

We adopt this approach for artificial evolution. By specifying tiepoints, sufficient control can be exerted about structures in 2-D images as to provide useful variants to the images being varied.

Let us look more closely at the operations usually implemented in EAs.

6.1 Population

One or more individual(s) of simulated animals or plants constitute(s) a population. Many groups containing one or more species are able to constitue one or more population(s). Structurally, the content of a simulated model is usually composed of components. In our example, a bird is composed of a body, bill, breast, wing, claw, leg, foot, tail, feather, etc. and a fish, tail, anal fin, dorsal fin, pelvic fin, scales, pectoral fin, gill, snout or mouth, eye, etc. The variation process can be applied to the entire model representing the whole structure, just might also be applied to its components. By using many tiepoints in a component, influence can be exerted to any necessary degree about the details of the evolving structure. The algorithm is achieved as follows:

initialize a population of individuals
initpopulation P (t);
evaluate fitness of all individuals in the first population
evaluate P (t);

6.2 Selection

The scientist can select any simulated individual, from any population, which subsequently reproduce with the variation process. The following is the selection algorithm:

select sub-population for offspring production
P′ := selectparents P (t);

Before presenting the entire interactive evolution system, we shortly discuss other representations of images. Procedural models of images can be characterized by certain parameters that must be interpreted in the appropriate context. The parameters constitute the genotype of an image. Its interpretation is the genotype-phenotype mapping and the resulting image is the phenotype. Because the number of structural elements usually varies from image to image, it is necessary to allow for variable-length genotypes.

Variation supports evolution to produce more offspring and to construct different body shapes. Variation takes place here on the level of parameters that are subjected to normally distributed random mutations as well as to intermediate or discrete recombination operations.

6.3 Recombination

Recombination is implemented as a global operation by which two models of simulated animals exchange information. We use as many tiepoints as necessary to conserve the underlying structure in two models "A" and "B". A recombination would then be quantified in the model space between "A" and "B" by a certain parameter indicating the degree of "intermediateness" of a variant. Figure 2 demonstrates the method of recombination. Different variants of fish are shown. Note that recombination always operates within the present generation. The recombination algorithm follows.

recombine the genotypes of selected parents
recombine P′ (t);

An algorithm for smoothly blending between two models supports the process of recombination between individuals. These models are composed of 2-D or 3-D polygonal shapes. The algorithm uses a physical model in which one of the shapes is considered to be created by wire, and the result is observed . The first shape can be streched or bent into the second shape. This process is known as metamorphosis, shape interpolation or shape blending. It has important application in the change and expansion in biological processes.

We can even recombine different kinds of simulated animals and get different kinds of descendants. .

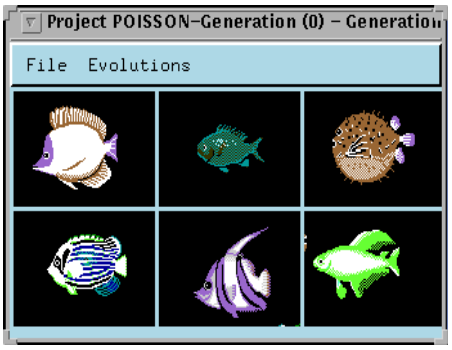

Figure 1: This figure shows the first generation of six individuals of fish forms

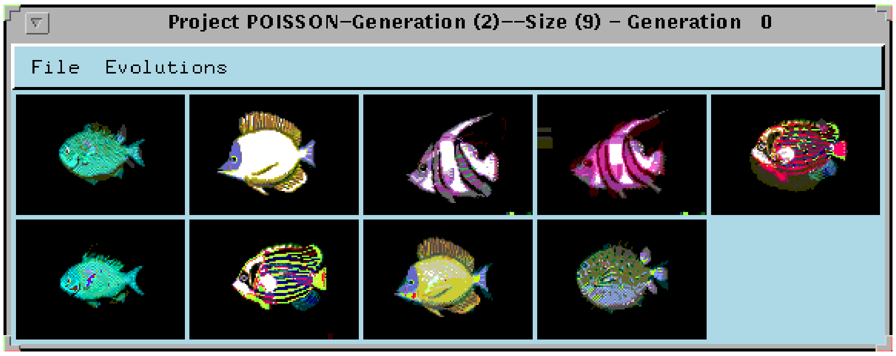

Figure 2: This figure shows an example recombination of fish forms after two generations

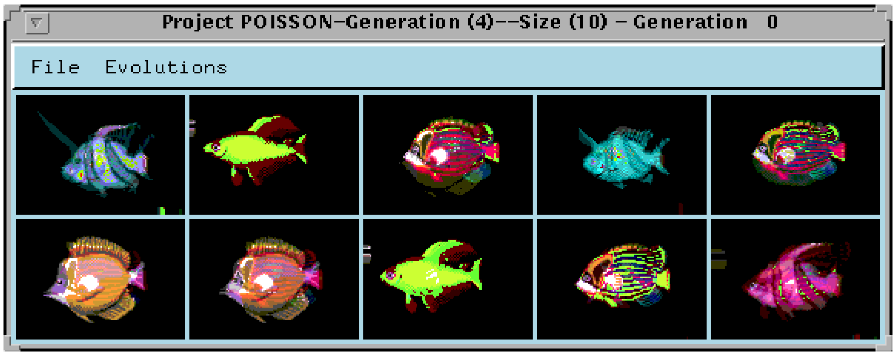

Figure 3: The ten fish forms are generated by recombination. After four generations

6.4 Mutation

Mutation is commonly considered to be a local operation that does not radically change the resulting phenotype. In order to provide this feature in bitmap image evolution, we use a very small number of tiepoints. A one-point mutation would select one tiepoint in a model. The corresponding tiepoint in a second model would then be used as a source for novelty, by providing information into which direction to evolve the original model. Structure is conserved because tiepoints in both models correspond to each other. A parameter would then be used to quantify the degree of substitution in the model. The following is the mutation algorithm:

perturb the breeded population randomly

mutate P′ (t);

Note that the second model, from which novelty is gained, is not necessarily in the present generation of the evolutionary process. Instead, a generation 0 of models, equipped with a number of tiepoints is used for mutation. By selecting one tiepoint in a model of generation n that corresponds to a tiepoint in a model of generation 0 and constraining the effect to a local neighborhood, we provide a path for evolving between the two models. The generation 0 model helps to form equivalence classes between structures expressed as tiepoints. Some domain knowledge must be used in the process of tiepoint selection for generation 0.

Figure 4 demonstrates the mutation process by using whale models. A sequence of local variations take place legged whales. The result is displayed in a transition form.

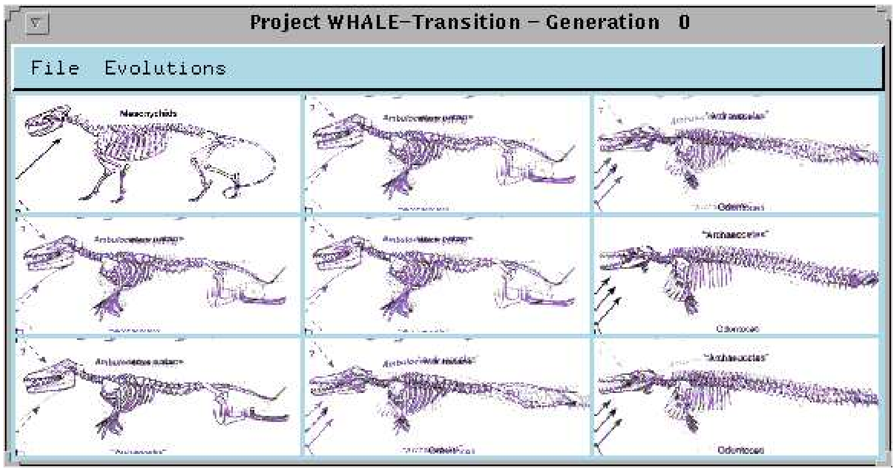

Figure 4: This figure shows the transition form of legged whales to leggless whales.

6.5 Reproduction

The resulting models reproduce to constitute the next generation. In each new generation, every individual gets its special genotype, which defines it. Figure 6 demonstrates the recombination and reproduction process. These genotypes are inherited from the previous generation. The reproduction algorithm:

evaluate it's new fitness

evaluate P′ (t);

select the descendants from actual fitness

P := survive P,P′ (t);

7 Jardin2 – A System for Simulated Evolution

Jardin2 is a digital image interpolation program, that allows the evolution of simulated models of plants and animals, and runs under the X Window System (5; 12; 31). Jardin2 loads and saves models populations. It provides facilities to store points in images, to evolve images, and to apply the evolutionary process. Points are inherited from generation to generation, with generation "0" provided by the user. With a very small population, between two and 20 models per generation and over a short time, a scientist can select new generations of simulated animals or plants models. This process is repeated until some statement individuals in the population are generated.

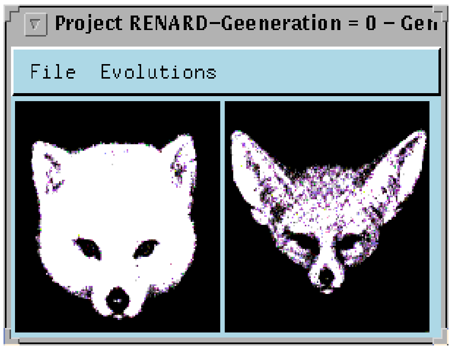

Figure 5: This figure shows two different fox parents, ice fox and desert fox

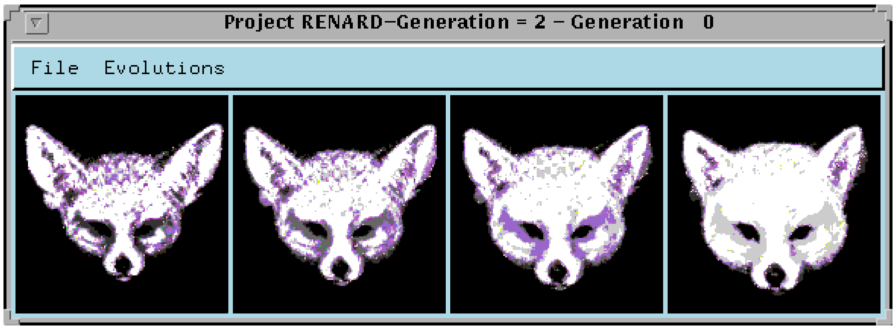

Figure 6: This figure shows the result of artificial recombination between an ice fox and a desert fox, resulting in different kinds of fox variations including shapes that actually exist in nature, such as red fox.

8 Summary and Conclusions

This paper demonstrates how interactive evolution might be applied to natural evolution models representation. The main idea is to combine the concepts from interactive evolutionary algorithms with the concepts of image interpolation from computer graphics. Structure in animal and plant models is substituted by a collection of tiepoints, manually provided by the user or automatically generated. Evolution then proceedsalong paths constrained by the set of tiepoints in all models. The original generation is kept as a source for mutations, which allows for new models to be created. In contrast, recombination always works on models of the present generation.

In our version of *interactive evolution*, the user selects individuals which then are reproduced to constitute the next generation. This technique can be applied to simulation and animation, for scientific visualisation, in the biological domain and like in paleontology and ontogony.

Acknowledgements

Funding from the German Bundesministerium für Forschung und Technologie (BMFT) under project EVOALG is gratefully acknowledged.

References

J. Arvo. *Graphics Gems*. Academic Press, California, 1991.

Th. Bäck. *Evolutionary Algorithms in Theory and Practice*. Doctoral dissertation, University of Dortmund, 1994.

W. Banzhaf. Genotype — Phenotype Mapping and Neutral Variation, A case study in Genetic Programming. In R. Männer Y. Davidor, H.P. Schwefel, editor, *Proc. PPSN III*, Jerusalem, 1994. Springer, Berlin, 322 - 332.

A. Berta. What is a whale? *Science 263*, pages 180–181, 1994.

E. Cutler, D. Gilly, and T. O'Reilly. *The X Window System in a Nutshell*. The Definitive Guides to the X Window System. O'Reilly & Associates, Inc., Sebastopol, CA, 1993.

L. Davis. *Handbook of Genetic Algorithms*. Van Nostrand Reinold, New York, 1991.

R. Dawkins. *The Blind Watchmaker*. Longman, Harlow, 1986.

D. B. Fogel, L. J. Fogel, W. Atmar, and G. B. Fogel. Hierarchic methods of evolutionary programming. In D. B. Fogel and W. Atmar, editors, *Proc. of the First Annual Conference on Evolutionary Programming*, La Jolla, CA, 1992. Evolutionary Programming Society, 175-182.

D.B. Fogel. *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. IEEE Press, New York, 1995.

L. J. Fogel, A. J. Owens, and M. J. Walsh. *Artificial Intelligence through Simulated Evolution*. Wiley, New York, 1966.

T. A. Foley. *Computer Graphics Principles and Practice*. Addison-Wesley, MA, 1992.

T. Gaskin. *DGXWS - PEXlib Programming Manual 3D Programming in X*. O'Reilly & Associates, Sebastopol, CA, 1992.

P.D. Gingerich, B.H. Smith, and E.L. Simons. Hind limb of eocene basilosaurus: evidence of feet in whales. *Science 249*, pages 154–156, 1990.

A. Glassner. *Graphics Gems*. Academic Press, California, 1990.

D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, MA, 1989.

W. D. Hillis. Co-evolving parasites improve simulated evolution as an optimization procedure. In *Physica D42, 228-234*, 1990.

J. H. Holland. *Adaption in Natural and Artificial Systems*. Ann Arbor, The University of Michigan Press, 1975.

D. Kirik. *Graphics Gems III*. Academic Press, California, 1992.

R. Lewontin. Adaptation. *Special Issue on Evolution*, pages 105–114, 1978.

E. Mayr. Evolution. *Special Issue on Evolution*, pages 2–13, 1978.

I. Rechenberg. *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann–Holzboog, Stuttgart, 1994.

D. Ruprecht. *Geometrische Deformationen als Werkzeug in der graphischen Datenverarbeitung*. Doctoral dissertation, University of Dortmund, 1994.

H.-P. Schwefel. *Numerical Optimization of Computer Models*. Wiley, NY, 1995.

K. Sims. Artificial evolution for computer graphics. *Computer Graphics*, (25), 1991.

J. Smith. Plants, fractals, and formal languages. *Computer Graphics, Vol.18, No.3, 1-10*, 1984.

J.G.M. Thewissen, S.T. Hussain, and M. Arif. Fossil evidence for the origin of aquatic locomotion in archaeocete whales. *Science 263*, pages 210–212, 1993.

S.P. Todd and W. Latham. *Mutator, a Subjective Human Interface for Evolution of Computer Sculptures*. IBM United Kingdom Scientific Center Report, 1991.

J. Valentine. The evolution of multicellular plants and animals. *Special Issue on Evolution*, pages 49–66, 1978.

A. Watt. *3D Computer Graphics*. Addison-Wesley, Reading, Massachusetts, second edition, 1993.

G. Woldberg. *Digital Image Warping*. IEEE Computer Society Press, Los Alamitos, CA, 1990.

D.A. Young and J.A. Pew. *The X Window System Programming & Applications with Xt*. Prentice Hall, Englewood Cliffs, NJ, 1992.