Generating Diverse Ethnic Groups with Genetic Algorithms

Tomas Trescak
IIIA, Artificial Intelligence Research Institute
Spanish Research Council
Barcelona, Spain
ttrescak@iiia.csic.es

Anton Bogdanovych, Simeon Simoff
MARCS Institute
University of Western Sydney
NSW, Australia
a.bogdanovych@uws.edu.au
s.simoff@uws.edu.au

Inmaculada Rodriguez
Applied Mathematics Department
University of Barcelona
Barcelona, Spain
inma@maia.ub.es

ABSTRACT

Simulating large crowds of virtual agents has become an important problem in virtual reality applications, video games, cinematography and training simulators. In this paper, we show how to achieve a high degree of appearance variation among individual 3D avatars in generated crowds through the use of genetic algorithms, while also manifesting unique characteristic features of a given population group. We show how virtual cities can be populated with diverse crowds of virtual agents that preserve their ethnic features, illustrate how our approach can be used to simulate full body avatar appearance, present a case study and analyze our results.

Categories and Subject Descriptors

H.5.1 [**Multimedia Information Systems**]: Artificial, augmented, and virtual realities

Keywords

Crowd Simulation, Genetic Algorithm, Avatar Appearance

1. INTRODUCTION

Virtual worlds and 3D games use 3D avatars[1] for user's physical representation in the virtual space as well as for simulating computer-controlled non-player characters. 3D avatars are also widely used in the movie industry, training simulators and health systems. In most instances, such avatars are manually designed, but often there are situations when such manual design is not practical. The most common case when avatar design automation is required is when a large crowd of computer-controlled avatars must be simulated to perform a particular group activity.

As an example of automatic crowd generation, one of the most popular solutions in the movie industry, is to utilize *Massive*[2] software that offers facilities for creating a given number of avatar clones and further modification of those clones by introducing a slight variation into their appearance features. This technique was widely used in Peter Jackson's The Lord Of The Rings film trilogy[3] for simulating battles.

The approach in generating crowds, taken by systems like Massive, is to use a number of manually created avatar shapes and randomly modify some shape parameters and textures to introduce the variety. To avoid non-plausible distortions of the resulting shapes, the features being changed are often limited to randomly selecting a clothing texture from a predefined list or modifying the height and width of the avatar. Thus, such systems are limited in terms of introducing a variety into crowds.

What is often desired in crowd simulation - is to have a diverse crowd with representatives of various genders and age groups, having different appearance, while still consistently maintaining the distinct features of their ethnic group. Simulating such diverse groups requires identifying the characteristic features that represent a given population and defining the acceptable range of variation in these features, as well as their intelligent manipulation. When generating crowds of avatars in an automatic fashion, without satisfying these conditions, either the believability of the crowd appearance or its diversity will be very limited.

Nowadays, most modern game engines and virtual worlds offer facilities to design parametric avatars [15], where avatars can be edited using hundreds of individual parameters (head shape, nose length, eye size, etc.). Such parametrisation offers enormous flexibility in generating unique avatar crowds and calls for revisiting existing crowd generation techniques.

Having parametric avatars allows to go beyond standard randomization techniques [18] for the crowd simulation. We suggest to introduce diversity into avatar appearance by mimicking genetic rules of reproduction present in nature. Under normal circumstances, humans and animals, when producing their offsprings, manage to achieve enough variety in the appearance of their children, while also preserving their distinct personal and ethnic characteristics, as well as ensuring that their body shape and facial features remain within the acceptable range of variation for the given species.

Thus, in this paper, we introduce an algorithm that generates visually unique avatars following the representations and techniques used in genetic algorithms theory [7] (e.g. crossover and mutation). This algorithm generates individuals that respect the visual, racial, cultural and behavioural

[1]3D avatar is an animated, emotive, complex model representing a user in a graphical form that ranges from actual resemblance of the human user to a talking fish or a robot.

[2]http://www.massivesoftware.com/ (last access 07/2012)
[3]http://www.lordoftherings.net/ (last access 07/2012)

features of a defined population as well as genetic inheritance of these features. Using genetic mutation, we add novelty to generated avatars. In addition to the visual aspects of crowd simulation, our approach is also applicable to generating unique personalities of virtual agents.

Our work focuses on generating large crowds, where each individual is a unique, distinguishable member of some ethnic group (see Figure 1). It should not be treated as an attempt to closely mimic genetic reproduction found in nature. Simulating the underlying processes behind forming body tissues and bone structure based on the DNA code is quite complicated, as those mechanisms are not yet fully understood and are not computationally feasible at present. Instead, our algorithm uses the same basic principles, but deals with a high-level representation of genetic code and a very straight-forward technique for manipulating it to produce the desired changes in the avatar appearance.

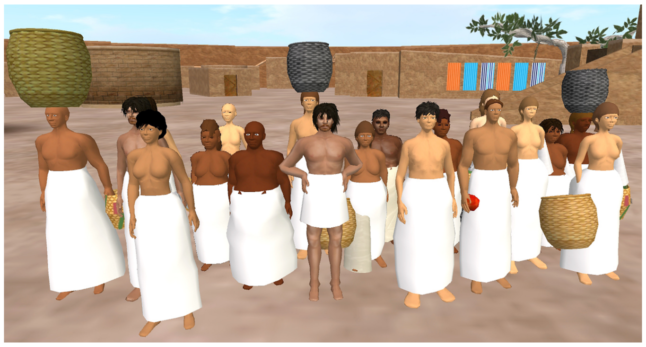

Figure 1: Generating Diverse Ethnic Groups

In our approach, we first isolate and quantify visual features of an avatar (e.g. nose width) and then use approaches and techniques from *genetic algorithms* rather than mimicking biological behavior. Genetic algorithms belong to the field of *evolutionary computing* where computational approaches in optimization seek inspiration from evolution and genetics.

Isolation and quantification of visual features of an avatar can be a tedious process, but by doing so we can achieve a great level of detail and variety in generated avatars [10]. For example, for heads of human avatars some can represent the craniofacial measures quantified by their minimal and maximal values [21, 20]. Limiting these features by an interval allows us to better control the appearance of an avatar, prohibiting the creation of unwanted (e.g. implausible) results. Also, we can combine them to represent physical properties (e.g. fatness), or even emotions (e.g. happiness). Another possibility is to define different intervals of quantification for visual features of children and adults.

A big advantage of isolating visual features is the reusability with different 3D models. It also becomes possible to manipulate non-human 3D models while using the same basic principles (e.g. manipulate a visual feature of eye-size in a 3D model of an animal).

When visual features are isolated and quantified, all their values for an individual are extracted to form the *chromosome*. A specific value of a visual feature is named *gene*. Genetic algorithms provide different operators that combine (i.e. crossover) and manipulate (i.e. mutation) parent chromosomes to generate their children with a defined level of novelty. Next, we explore the related work in avatar generation, describe our approach and discuss experimental results.

2. RELATED WORK

Avatar generation and crowd generation methods vary in how to model a single individual and in the approach of making every crowd member unique. In one of the first attempts to generate a population of unique 3D characters, [4] created a system that generated facial models. The model was based on randomization of anthropometric measures applied to B-spline surfaces. Later, [2] used Principal Component Analysis (PCA) to analyze datasets of facial features to extract base vectors from the face, and used these vectors to generate new, unique faces. This work was extended by [1] to generate the whole body of an avatar. A different approach was taken by [12] and [18] who used variance of attachments and textures over a predefined set of avatars, where avatars mainly differed in size and the type of textures, while still appearing as clones that undergone a minor modification.

In other works, authors considered different means of crowd generation and isolated specific visual parameters (i.e. body and clothing) which deform related 3D models. The values of these parameters can be randomized in order to produce unique crowd members [16] [11].

Identification of such visual parameters inspired researchers to encode their values into genetic structures and apply genetic algorithms theory to generate unique crowd members. Ventrella [19] was one of the first to discuss the possibilities of storing and modifying the avatar properties in a “chromosome”, represented by an array of integer values and then modifying those properties to generate a 2D sketch of an avatar with different appearances. Technological details of this work, however, were not explained and the main focus was limited to generating a single avatar, not a crowd [19].

While Ventrella worked with 2D sketches a similar approach was taken by [10] to generate 3D avatars. Authors in [10] presented a method for encoding a rigged 3D avatar into a list of bones and their dimensions with further modification of those using genetic principles. The resulting avatar generation method often produced unrealistic body shapes and was intended to be used in a supervised manner, where a user was presented with a large list of possible avatars and was expected to manually select the most suitable one.

The most similar work to our approach was done by [21, 20], who investigated generating an avatar’s appearance using genetic inheritance principles. Genetic inheritance was approached from the biological perspective, where each child chromosome holds a copy of mother’s and father’s chromosome. During the reproduction process, child chromosomes were duplicated, combined and then split into four ”gametes”. Through the process called “fecundation” a father gamete and a mother gamete are combined to produce the new child’s chromosomes. The parents’ gene values from the chromosome are combined to visualize the final value. This method relied on direct modification of 3D meshes and was limited in the number of parameters that can be modified, and similar to [10] often produced non-plausible results.

We advance existing work by applying genetic algorithms rather than mimicking biological evolution like [21]. Similar to [10], we encode visual properties as genes, which form chromosomes, but rather than using meshes or bones - we rely on fully parametric avatars. We use genetic operators to generate new, unique individuals, which due to parametric nature will be limited to plausible characters. Our contributions include the definition of new techniques applied during replication, such as deep inheritance, which provides the possibility of inheritance of features from indirect ancestors. Another contribution is the definition of genotype rules, which can define key ethnic features that have to be preserved during replication, or dependencies between visual features, e.g. making an avatar fat. As a result, we can generate ethnic crowds with a high degree of variation across hundreds of body features, while also maintaining the similarity with avatar ancestors in a classical genetic sense. This allows us to simulate avatar groups of unlimited size, where every individual has a unique appearance, while at the same, is identifiable as the member of its ethnic group based on characteristic appearance features.

3. AVATAR GENERATION

In this section, we propose a general-purpose model for automatic generation of 3D avatars, which can be deployed in modern video games or virtual worlds. This model is not limited to human avatars and can be used with other avatar types (e.g. humans, animals, fishes, robots, orcs) as long as they have distinguishable *visual features*. Next, we focus on generating individual virtual agents with unique appearance from a small sample of manually designed avatars.

3.1 Genetic algorithms

Genetic algorithms (GA) belong to the larger class of evolutionary algorithms, which generate solutions to optimization problems using techniques inspired by natural evolution, such as *inheritance*, *mutation*, *selection*, and *crossover*.

In the history of evolutionary computing, four important paradigms served as a base activity of the field: genetic algorithms [7], genetic programming [9], evolutionary strategies [14] and evolutionary programming [5]. Their differences lie within the terminology behind the algorithms, the reproduction operators and selection methods. In our work, we use the terminology and procedures from genetic algorithms.

A traditional genetic algorithm defines and manipulates individuals at the level of their *chromosomes*, where a chromosome is represented as a *fixed-length bit string*. Each position in the string is assumed to represent a particular feature of an individual, called *gene*. Usually, the string is "evaluated as a collection of structural features of an individual, which have little or no interactions" [17]. To produce a new generation of individuals, we combine parent's chromosomes using a *crossover* operator. Combining parent's genes allows gene *inheritance*. To introduce novelty in the population, we apply *mutation* to the current chromosome.

Genetic algorithms are employed in numerous fields, such as *search*, *optimization* and *machine learning* [6]. In most of these cases GA are used as a search heuristic that mimics the process of natural evolution. In this approach, the GA are treated as an optimization algorithm utilizing a fitness function to *select* a partial solution of the original problem, till reaching some predefined threshold for suboptimal solution.

In our approach, we are not using GA as an optimization algorithm. This is why we do not use the *selection* and focus only on *crossover*, *mutation* and *inheritance*. In the following sections, we provide the formal representation of genetic data and operators. Then we contemplate our design of deep inheritance, also called gene skipping, and explain the generation algorithm.

3.2 Formal Representation of Genetic Data

In this section, we formalize genetic concepts used during the avatar generation process. First, we define visual information or definable visual traits of an avatar. For the purpose of generation of unique avatars such information needs to: (i) express the specific appearance trait of an individual and (ii) must be quantifiable. Therefore, this information must hold the quantifiable visual descriptors of an individual. For example, "height" visual descriptor can be quantifiable by a numeric interval, from 0 to 100, with 0 being short to 100 being tall (respecting the natural height of humans). Moreover, a visual descriptor can quantify the shape, texture or color of an avatar's body part. We define the visual descriptor as the *visual feature* (see Table 1):

Definition 1. *A* **visual feature** holds a quantifiable descriptor of a body part of an avatar. It is defined as $vf = \{$Id, Name, Value, Interval, Minimum, Maximum$\}$ where:

1. *Id* is a unique integer identifier, $id \in \mathbb{I}$
2. *Name* is a string that identifies the visual feature, $name \in \mathbb{S}$
3. *Interval* defines the range of visual feature values with textual descriptor of *Minimum* value (e.g. short) and *Maximum* value (e.g. tall).

Id	Name	Interval	Minimum	Maximum	Gene
33	Height	[0,100]	Short	Tall	81
45	Nose Size	[0,100]	Small	Big	21
67	Arm Length	[0,100]	Short	Long	27
80	Gender	{0,1}	Male	Female	1

Table 1: Examples of definitions of visual features

Using our approach, each avatar is first described by a list of *visual features*. These features should thoroughly characterize its appearance. When such list is defined, we quantify the values of visual features into *genes*. Genes are real or boolean values of visual features, which allow us to perform genetic operations, such as crossover and mutation.

Definition 2. A **gene** $g \in \mathbb{R} \cup \mathbb{B}$ represents a real or boolean value of the *visual feature* $g = vf_{value}$ (i) to quantify the *presence* of the visual feature, $g \in \mathbb{B}$; (ii) to quantify the *strength* or a *position* of a feature (e.g. height with values close to minimum representing short avatar and values close to maximum representing a tall avatar), $g \in \mathbb{R}$.

The ordered set of avatar's genes is called *chromosome*. A chromosome holds the complete genetic information of an individual. Order of the genes in a chromosome is specified by *genomic sequence*.

Definition 3. A **genomic sequence** defines the genetic composition of a chromosome of a specific group, forming a representative gene ordering of a species or group. A genomic sequence GS for a chromosome of length n is a set of indexes which define the ordering of genes in the chromosome: $GS = \{i_1, i_2 \ldots i_n\}$.

Definition 4. A **chromosome** of length n, c_n, is an ordered list of n genes $g_1, g_2 \dots g_n$, where gene order is given by a genomic sequence GS, that is $c_n = \{g_{i_1}, g_{i_2} \dots g_{i_n}\}$ where $i_1, i_2 \dots i_n \in GS$.

A chromosome identifies all visual features of an individual; thus, it is possible to analyze chromosomes of an existing population and identify the genes that are common for a specific ethnic group. These genes are marked as ethnic-specific and they are modified only steadily during the reproduction, in order to maintain the ethnic-specific features.

Furthermore, to operate with the genetic inheritance in population we store all the parent-child relationship information in a *genealogy tree*. As a result, it allows us to navigate within the relationships of the population. This tree provides information on ancestors or siblings of the generated avatar and provides access to their chromosomes. Its name comes from the graphical representation where the family ancestors are visualized in the tree-like structure, also called as *family tree*. A *genealogy tree* is usually represented as a directed acyclic graph (see Figure 4).

3.3 Formal Definition of Genetic Operators

In our approach, we first define the initial population, that characterizes the base groups of population. Members of these groups have distinguishable visual features that define such group (e.g. asians with asian eyes, africans with african skin). Then, we use this characteristic information to create new individuals. Therefore, we may refer to our generation process as to *reproduction*. During reproduction, we use the information stored in *chromosomes* of parents and combine and modify them using a *genetic operator* to reproduce chromosomes of new unique individuals.

In the previous section, we have formalized genetic concepts related to the reproduction process, that is *genes*, *chromosome*, *genomic sequence* and *genealogy tree*. In the following sections, we define all the genetic functions and operators. Such operators are responsible for *crossover*, *mutation* and *inheritance* of genetic information from parents. Moreover, we introduce the *genotype rule* a mechanism to explicitly control the generation process, preserve the group characteristic visual features and introduce relationships between genes. Our algorithm first combines parent chromosomes using the *crossover* operator and then uses *gene skipping* to inherit some genes from its deeper ancestors. Next, to introduce novelty, it *mutates* the produced chromosome. Finally, using a set of predefined *genotype rules*, it adjusts this chromosome to respect the population group properties.

3.4 Crossover

Crossover is a genetic algorithm operator that is used to vary and combine the genetic information stored in chromosomes from one generation to the next one [17]. Analogous processes from biology are *genetic reproduction* and *biological crossover*. In this process, a crossover function selects and combines the genes from one or more parents to create a new child chromosome. We formalize different crossover operators. Some operators (i.e. clone, split) are traditional, coming from the genetic algorithm theory, some are defined by us to meet our needs (i.e. exchange, fuzzy). The first traditional crossover operator is *clone* operator, which copies the genetic information directly from one of the parents.

Definition 5. Given the mother chromosome c^m consisting of genes $c^m = g_1^m g_2^m \dots g_n^m$ and the father chromosome c^f consisting of genes $c^f = g_1^f g_2^f \dots g_n^f$ a **clone** operator $\oplus^q : C \times C \to C$ is defined as $c^m \oplus^q c^f = g_1^q g_2^q \dots g_n^q$ where q is either denoting mother (m) or father (f) genes.

Cloning is the simplest and fastest crossover technique, but it does not provide many functionalities in mixing genetic information from parents. The simplest crossover operator that allows us to combine this genetic information is the *split* operator (in GA terminology also known as one-point crossover [17]). Figure 2 contemplates the operation of the split operator. This technique splits the chromosome after a specific gene, called *split point*, and all genes on the left of this split point are copied from one parent, and all genes on right are copied from the other one. The split point specifies a *father-mother ratio*, r^{fm}, meaning what percentage of genes should be copied from mother and which from father. The split technique is applicable to more parents, where we select one split point for each additional parent. Formally:

Definition 6. Given the mother chromosome c^m consisting of genes $c^m = g_1^m g_2^m \dots g_n^m$ and the father chromosome c^f consisting of genes $c^f = g_1^f g_2^f \dots g_n^f$ and the split point $0 \leq s \leq n$, a **split** operator $\ominus : C \times C \to C$ is defined as $c^m \ominus c^f = g_1^m g_2^m \dots g_s^m g_{s+1}^f \dots g_n^f$

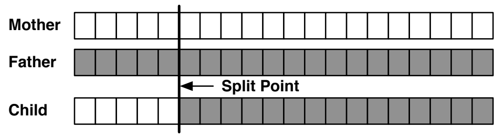

Figure 2: Crossover technique called "split"

Previous crossover operators allow none, or very limited mechanisms for combining and experimenting with genetic information from parents. A flexible crossover operator, is the *exchange operator*. This operator combines parents' chromosomes by randomly selecting two complementary subsets of genes from parents and joining them. Formally:

Definition 7. Given the mother chromosome c^m consisting of genes $c^m = g_1^m g_2^m \dots g_n^m$, the father chromosome c^f consisting of genes $c^f = g_1^f g_2^f \dots g_n^f$ and the exchange selection function $e : G \times G \to G$ that for input father and mother genes outputs only one of them, we define a crossover **exchange** operator $\otimes : C \times C \to C$ as $c^m \otimes c^f = e(g_1^m, g_1^f) \cdot e(g_2^m, g_2^f) \dots e(g_n^m, g_n^f)$.

The process of gene exchange is shown in Figure 3, clearly marking genes selected from mother and from father. The number of genes selected from the father and mother is controlled by the *father-mother ratio*.

For our purposes, we define a new crossover operator named *probabilistic fuzzy operator*. This operator is similar to *exchange* operator, but rather to copy the exact values of parent genes it combines values in the interval given by the values of genes from the father and mother. This operator brings even more variety into the generation process. We use this operator to evaluate our approach in Section 4.

Definition 8. Given the mother chromosome c^m consisting of genes $c^m = g_1^m g_2^m \dots g_n^m$, the father chromosome c^f

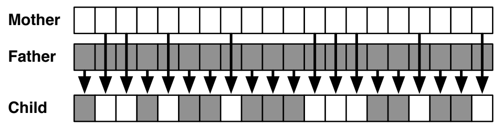

Figure 3: "Gene exchange" crossover technique

consisting of genes $c^f = g^f_1 g^f_2 \dots g^f_n$, the parent *gene selector function* $s^i_{rfm} : 2^G \rightarrow \{0,1\}$ which for position i, where $0 \leq i \leq n$, selects either mother or father gene depending on probability given by the father-mother ratio r^{fm} and the *fuzzy function* $f : \mathbb{I} \rightarrow \mathbb{R}$ which for gene on position i selects a random value in the interval given by $f(i) = [s(i), (g^m_i - g^f_i)/2]$, we define a **fuzzy** crossover operator $\oslash : C \times C \rightarrow C$ as $c^m \oslash c^f = f(1) \cdot f(2) \dots f(n)$.

We can define even more crossover operators, but for our purposes, the split, exchange and fuzzy are enough. An example of custom crossover operator would take the arithmetic average of mother and father gene values.

3.5 Inheritance and Gene Skipping

Looking at ourselves in the mirror and analyzing our visual appearance, most probably we find visual traits from our parents. Nevertheless, many of our traits go even deeper into our ancestry tree, and sometimes we look more similar to our grandparents than to parents. Inheriting visual and behavioral features from our predecessors is possible due to a process called *gene skipping*. Although, sometimes our visual features have little similarity with our ancestors and are the result of *mutation* or are altered by some external factor (see Section 3.6).

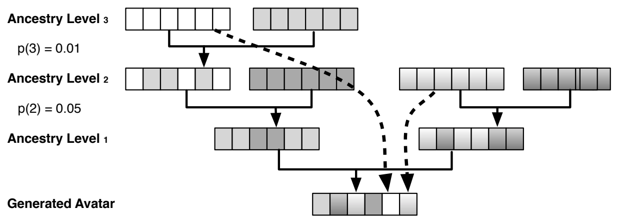

Figure 4: "Gene skipping", where $p(n)$ denotes the probability of inheriting gene from ancestry level n

Gene skipping is a natural genetic process that occurs during the reproduction process, when we obtain the genetic information from our deeper ancestors. Such genetic information does not only alter our appearance, but also behavior, personality and predispositions.

Figure 4 contemplates the process of gene skipping. In our approach, this process becomes active after combining chromosomes using a crossover operator. For each gene in the chromosome, we evaluate the possibility of inheriting this gene from our ancestors. The probability value is bound to an *ancestry level*. Ancestry level of our parents is 1. Our grandparents are our second closest ancestors, their level is 2, great-grandparents have level 3 and so on. For a given gene, we define the probability of skipping it from a predecessor in ancestry level n as a quadratic function $p(n) = \frac{1}{5 \times 2^n}$, where $n \geq 2$. We have not found unified and exact values of such probabilities in the literature; therefore, we have modelled it to reflect the quadratic decrease with relation to older ancestors. In this manner, there is a 5% probability of inheriting a given gene from our grandparents, 2.5% from our great-grandparents, 1.25% from great-great-grandparents. The gene skipping technique introduces an interesting level of variability of avatar appearance. Next, we present another genetic operator called *mutation*, which further extends the variability of generated results.

3.6 Mutation

In genetic algorithms, mutation is a genetic operator used to provide diversity from one generation to the next. Mutation alters one or more gene values in a chromosome from its initial state. Using mutation, the solution may change entirely from the previous solution. Hence a genetic algorithm can come to improved solutions using mutation [6].

We use mutation to bring more variability to generated avatars, allowing them to have new visual traits, unknown to their ancestors. In our case, mutation is performed by randomly selecting a specific number of genes and modifying their values. The new value is either random, or taken from a specific interval (e.g. ± 20% of parent value). The mutation level is expressed by the percentage of the mutated genes from the total number of genes. In the bottom of the Figure 7, we see a woman avatar, PintoLae2 that is significantly different from his parents due to the high level of mutation set to generate this avatar.

3.7 Genotype Rules

In genetics, the *genotype* is a genetic makeup of a cell, an organism, or an individual. In another words, it is a measurement of how an individual differs within a group of individuals or species. We have borrowed this specific term for a mechanism that allows us to define the characteristics of individuals or population groups, using *genotype rules*.

We propose *genotype rules* to provide precise control over values of individual genes during the reproduction process. It is a powerful mechanism, that allows us to define the characteristics of a population group and to maintain their significant traits (e.g. asian eyes). Moreover, genotype rules can specify the relationship between genes. Such rule can, for example, force the inheritance of a group of genes only from one parent, depending on the gender of the generated avatar. Further we illustrate it on an example of generating female avatars with "ideal" breast-waist-hip proportions.

A genotype rule modifies the gene value of the reproduced chromosome depending on the other gene values stored in this reproduced chromosome, as well as gene value stored in any of its ancestors. Genotype rules use the *selection function* that allows them to select the gene value from its own chromosome or from a chromosome of a specific relative. The input of this selection function for gene n is a string, which encodes the full path to the given individual.

Definition 9. A **gene selector function** $s_g : \mathbb{S}^* \rightarrow \mathbb{R}^*$ is a recursive function which returns the value of a specific gene g from one or several relatives. Relatives are selected depending on the *input string* s in the form $s = X_1 \cdot X_2 \cdot \dots \cdot X_m$ where $X_{k|k \leq m} \in \{self, parent, child, sibling\}^*$. Values of X_k represent standard tree node selection functions and their concatenation specifies the path to specific relatives.

Examples of *input strings* S^*, to the gene selector function follow the subscript notation (i.e. $selectionFunction_{geneName}$) with the gene name $geneName$ of some selected individual appearing under the selection function name $selectionFunction$:

Avatar: $self$
Father: $parent_{gender}$ = 'male'
All my brothers that are taller than me:
$sibling_{gender}$ = 'male' $\wedge$ $sibling_{height} > self_{height}$

Definition 10. A **genotype rule** for gene g_i is a function $r_{g_i} : S_1 \times S_2 \times \ldots S_n \rightarrow \mathbb{R}$ which for a given input $s_1, s_2 \ldots s_n$, where $s_{k|k \leq n}$ is a gene selector function, returns the value of gene g_i.

Definition 11. A **genotype** is a set of genotype rules that characterizes a specific population group or an individual.

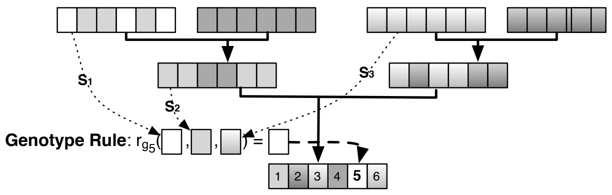

Figure 5: Selection process for rule $r_{g_5} : S_1 \times S_2 \times S_3$

Figure 5 contemplates the process of selection of genes for the genotype rule evaluation. Next, we show an example of how the genotype is used to characterize a population group and to specify a gene relation. For clarity, we do not use the number index of a gene r_{g_i} but the name of related visual feature r_{gender} that represents gene g_i.

Example: Ideal ratio
In this example we generate female avatars with breast-waist-hip ratio of (90-60-90). These dimensions are specified with four rules, where the diversity of generated values are respected by using them as measure to the new "ideal" proportions.

1. r_{gender} = "female"
2. $r_{breasts} = \frac{3}{8}(s_{breats}(self) + s_{waist}(self) + s_{hip}(self))$
3. $r_{waist} = \frac{2}{8}(s_{breats}(self) + s_{waist}(self) + s_{hip}(self))$
4. $r_{hip} = \frac{3}{8}(s_{breats}(self) + s_{waist}(self) + s_{hip}(self))$

The first rule defines that all generated avatars will be female. The remaining rules encode that breasts should be $\frac{3}{8}$ (same as $\frac{90}{240}$) of the ratio and hip $\frac{2}{8}$ of the ratio and waist $\frac{3}{8}$ of the ratio. This example shows how easily we can model relations between genes.

3.8 Algorithm

Algorithm 1 uses all the mechanisms introduced in the previous section. First, it combines the chromosomes from *father* and *mother*, using a specific *crossover operator* and creates a new child *chromosome*. Second, if allowed, performs a *gene skipping* using the new chromosome and information stored in the *genealogy tree*. Gene-skipping uses pre-defined probabilities to decide if a given gene should be skipped. Third, the algorithm *mutates* a specific amount of genes in the new chromosome. This amount is set by the *mutation level*. At last, the chromosome is modified according to *genotype rules*. The algorithm repeatedly executes all rules till no change in chromosome is detected. The generated chromosome contains all the information about visual appearance of a new unique 3D avatar. The values of genes of generated chromosomes are used to render the new avatar.

Algorithm 1: Algorithm for generating an avatar

```
Input: genealogyTree, mother, father, crossover
       operator, fatherMotherRatio, mutationLevel,
       skipGenes, genotype rules
Output: Chromosome representing a new unique 3D
        avatar
begin
    // combine parent chromosomes
    chromosome ← Crossover (crossoverType,
    fatherChromosome, motherChromosome,
    fatherMotherRatio);
    // gene skipping
    if skipGenes then
        chromosome ← SkipGenes (genealogyTree,
        chromosome);
    // mutate chromosome
    chromosome ← Mutate (chromosome, mutationLevel);
    // adjust chromosome according to genotype
    // execute the rules till no change is performed
    repeat
        adjusted ← false;
        foreach (rule ∈ genotype) do
            adjusted = adjusted ∨ ExecuteRule (rule,
            chromosome);
    until adjusted = false;
    return chromosome
```

Figure 6 shows the interface of the Genetic Mixer tool[4], that implements all the features of the presented Algorithm 1. We used the Genetic Mixer to generate examples in Figure 7.

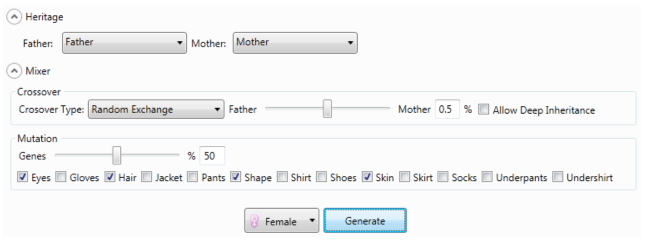
Heritage
Father: Father
Mother: Mother
Mixer
Crossover
Crosover Type: Random Exchange
Father — Mother 0.5 %
☐ Allow Deep Inheritance
Mutation
Genes % 50
☑ Eyes ☐ Gloves ☑ Hair ☐ Jacket ☐ Pants ☑ Shape ☐ Shirt ☐ Shoes ☑ Skin ☐ Skirt ☐ Socks ☐ Underpants ☐ Undershirt

Figure 6: Interface of the Genetic Mixer tool

Next, we evaluate the variability of the generated avatars and the performance of the algorithm.

4. EVALUATION

We have evaluated the generation algorithm on the diversity of the resulting avatar crowd depending on different

[4]see supporting video at http://youtu.be/Re7oVUFGis4

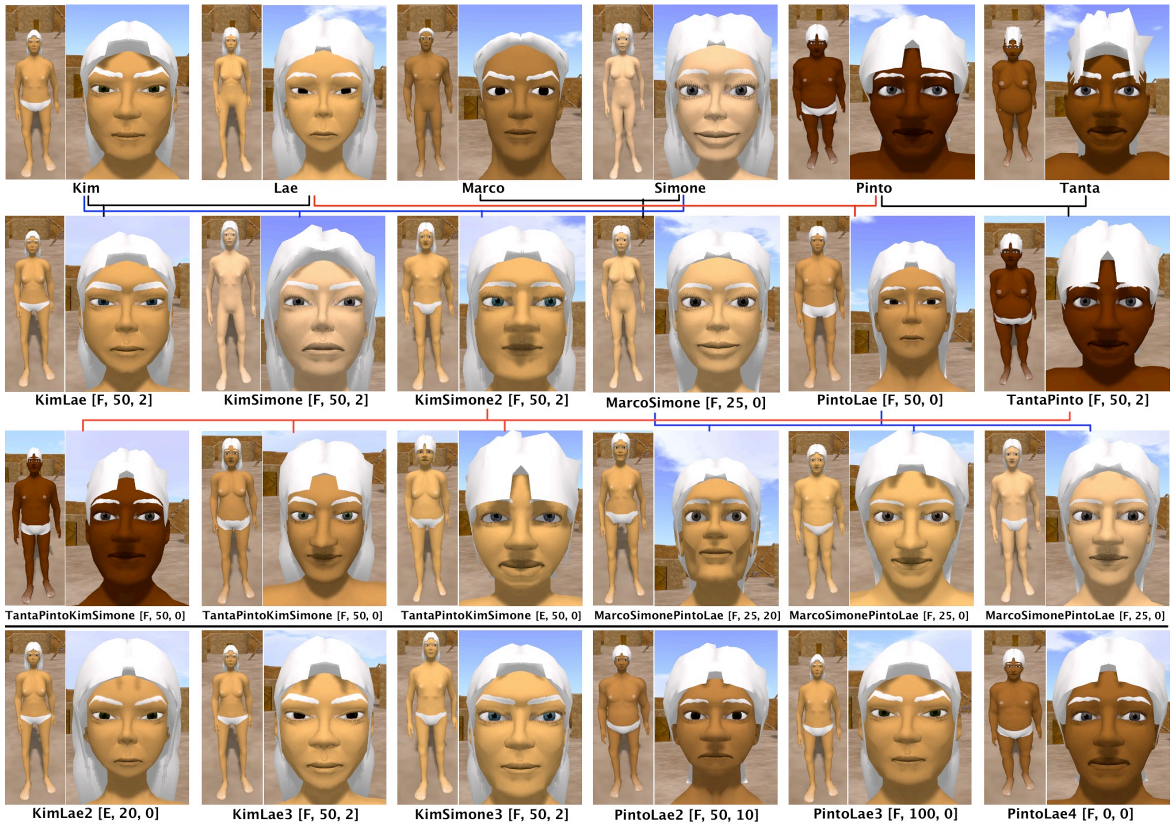

Figure 7: Avatars generated using our method. The top row forms the start population. The label of every figure contains following information: *Name [crossover, father-mother ratio, mutation level]*

input parameters of the algorithm, that is crossover operator, father-mother ratio and mutation level. We have also tested if characteristic ethnic features of individuals are preserved within their children. Some members of the resulting generated crowd are shown in Figure 7.

For our purposes, we have manually designed six avatars, three females and three males that form the base population of our virtual world. Two avatars are asian (Kim and Lae), two are african (Pinto and Tanta), one caucasian (Simone) and one arab (Marco).

To highlight our approach to identifying characteristic appearance features - we have designed the base population so that every avatar has white hair colour. One of the hypotheses that was tested during evaluation is that our algorithm will correctly identify this appearance feature as being characteristic for the given ethnicity and no or little mutation will be performed in relation to this feature, so that it is preserved in every member of the generated crowd.

We have isolated 200 visual features of each avatar, that are provided by OpenSimulator[5] and that define shape, skin, eyes and hair. Each of these visual features represents a gene in an avatar's chromosome. For the generation, we have used two crossover operators, *fuzzy* (marked as F in the avatar captions) and *exchange* (marked as E in the avatar caption). We have used different father-mother ratios and different mutation level. Each generated child is named by combining the father's and mother's name (e.g. child of Kim and Lae is named KimLae).

In the first row of Figure 7, we see all six avatars from our base population. In the second row, we see their children. Our system was able to correctly identify hair color as an important characteristic feature of the generated ethnicity, so all the generated avatars have white hair. KimLae was generated using the *fuzzy* with the ratio set to 50%. We can clearly see the resemblance from both father and mother. She has her father's eyes, but her mother's chin. Nose width is somewhere between father and mother. We can clearly see that she is asian. The same parameters were set for TantaPinto and we can see that she is black, resembling both parents. If we now focus on KimSimone and her brother KimSimone2, we see that they are both generated using the same parameters with significantly different results. This is due to the random nature of fuzzy operator and the use of *mutation*, which was set to 2%. When we look at the grand-kids of the base population avatars we can see how their visual features are combined.

In the last row we present a selection of avatars that we generated during the generation of 3000 avatars from our base population. PintoLae2 is significantly different from others due to the high level of mutation set to 10%. It's evident from the results that the algorithm generates a large variety of visually acceptable avatars respecting the genetic inheritance of ancestor features. Although, we were able to generate 3000 avatars, we are unable to display them all at

[5]http://www.opensimulator.org (last access 09/2012)

once, as we are limited by the capabilities of the OpenSimulator allowing to host a maximum of 100 avatars, but recent research on using Distributed Scene Graphs (DSG) [8] allowes simultaneous participation of thousands of users.

We have also measured the average speed of generation of one avatar during the generation of crowd of avatars. The speed of the algorithm linearly depends on the length of the chromosome, that is its time complexity for m agents is $O(m \times n)$ with respect to the length of a chromosome n. We have generated 3000 different avatars, using the most complex *fuzzy* operator with mutation level set to 2% and on average the algorithm generated a new avatar in 305 ms. The generation was performed on a MacBook Pro with 2.6 GHz Intel Core 2 Duo processor and 4GB of RAM.

5. CONCLUSION AND DISCUSSION

We have presented an algorithm for generating a diverse ethnic group of unlimited size from a small sample of manually designed 3D avatars. Genetic principles of inheritance, crossover and mutation are applied to the sample population to create new visually unique group members. Predefined genotype rules allow us to preserve important appearance features that are characteristic for the given ethnicity.

Generating large ethnic groups is important in domains like virtual heritage [3], where not only avatars must look different, but also have to appear as members of the same culture. In virtual heritage it is often needed to interact with individual avatars and inspect them at close proximity, so the use of cloning or colour substitution (as in [13]) is not sufficient as it is likely to be noticed. Also, introducing diversity by using a fully random approach to modifying individual parameters (as in [10]) may result malformed avatars and would modify the important appearance features that define the ethnicity of the simulated group.

It is important to mention that in simulating ethnic groups it is undesired for generated agents to have very close resemblance with their "genetic parents". Thus, the proposed genetic algorithms approach is also more beneficial than a biological approach of [21] as it results increased diversity, more transparent execution and better control over results.

The key drawback of our method is that it requires to have avatars defined in parametric form and to allow for quick modification of individual avatar parameters on the fly. However, many modern game engines, virtual worlds as well as popular 3D modelling packages support parametric avatars and provide the corresponding visual tools.

Future work includes extending this approach to generating agents with diverse personalities, as well as conducting an advanced study similar to [13], where the diversity of the generated avatars, resemblance with ancestors and preservation of ethnicity are evaluated in a more formal manner.

6. ACKNOWLEDGMENTS

This work was supported by the Digital Humanities Research Program of School of Computing, Engineering and Mathematics in the University of Western Sydney, as well as Consolider (CSD2007-002), TIN2009-14702-C02-01/02 and TIN2011-24220 Spanish Research Projects.

7. REFERENCES

[1] B. Allen, B. Curless, and Z. Popović. The space of human body shapes: reconstruction and parameterization from range scans. In *SIGGRAPH 2003 Papers*, pages 587–594, New York, NY, USA, 2003. ACM.
[2] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *SIGGRAPH 2009 Papers*, pages 187–194, New York, NY, USA, 1999. ACM.
[3] A. Bogdanovych, K. Ijaz, and S. Simoff. The City of Uruk: Teaching Ancient History in a Virtual World. In *IVA 2012 Conference*, pages 28–35. Springer, 2012.
[4] D. DeCarlo, D. Metaxas, and M. Stone. An anthropometric face model using variational techniques. In *SIGGRAPH 1998 Papers*, pages 67–74, New York, NY, USA, 1998. ACM.
[5] D. Fogel. *Evolutionary Computation*. IEEE Press, New York, 1995.
[6] D. Goldberg. *Genetic algorithms in search, optimization, and machine learning*. Addison-wesley, 1989.
[7] J. H. Holland. *Adaptation in natural and artificial systems*. MIT Press, Cambridge, MA, USA, 1975.
[8] Intel. Intel increases opensim avatar capacity 20 fold. http://www.hypergridbusiness.com/2011/06/intel-increases-opensim-avatar-capacity-20-fold/.
[9] J. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, 1992.
[10] M. Lewis. Evolving human figure geometry. Technical report, Ohio State University, 2000.
[11] N. Magnenat-Thalmann, H. Seo, and F. Cordier. Automatic modeling of virtual humans and body clothing. *Journal of Computer Science and Technology*, 19(5):575–584, 2004.
[12] J. Maim, B. Yersin, and D. Thalmann. Unique character instances for crowds. *Computer Graphics and Applications, IEEE*, 29(6):82–90, 2009.
[13] R. McDonnell, M. Larkin, B. Hernández, I. Rudomin, and C. O'Sullivan. Eye-catching crowds: saliency based selective variation. *ACM Trans. Graph.*, 28(3):55:1–55:10, July 2009.
[14] I. Rechenberg. *Evolutionsstrategie. Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann Holzboog, 1973.
[15] S. Robbins and M. Bell. *Second Life For Dummies*, chapter 5.2: Editing Your Avatar's Appearance. Wiley, 2008.
[16] H. Seo and N. Magnenat-Thalmann. An automatic modeling of human bodies from sizing parameters. In *Proceedings of the 2003 symposium on Interactive 3D graphics (I3D'03)*. ACM, 2003.
[17] S. Sivanandam and S. Deepa. *Introduction to genetic algorithms*. Springer Verlag, 2007.
[18] D. Thalmann. *Crowd simulation*. Wiley Online Library, 2007.
[19] J. Ventrella. Avatar Physics and Genetics. In *VW '00: Proceedings of the Second International Conference on Virtual Worlds*. Springer-Verlag, 07 2000.
[20] R. C. C. Vieira, C. A. Vidal, and J. B. Cavalcante-Neto. Reproducing virtual characters. *Computers and Graphics*, 36(2):80 – 91, 2012.
[21] R. C. C. Vieira, C. A. Vidal, and J. B. C. Neto. Simulation of genetic inheritance in the generation of virtual characters. In *Proceedings of Virtual Reality (VR'10) Conference*, pages 119–126. IEEE, 2010.