TOWARDS A NEW APPROACH FOR INTEGRATING MULTIMODAL USER INPUT BASED ON EVOLUTIONARY COMPUTATION

Frank Althoff, Marc Al-Hames, Gregor McGlaun, Manfred Lang

Institute for Human-Machine-Communication
Technical University of Munich (TUM)
Arcisstr. 21, 80290 Munich, Germany
email: {althoff, alhames, mcglaun, lang }@ei.tum.de

ABSTRACT

Multimodal interfaces provide flexible, intuitive, and error-robust interaction with complex information systems. In this work we describe an innovative statistical approach for combining multimodal user input that is based on principles adopted from evolution theory. A population of individuals, each representing a solution to the integration problem, compete for an optimal interpretation of the user interactions. Specially designed genetic operators recombine various characteristics of these solutions. The fitness of a single individual, measuring the certainty and the confidence of an integration result, is calculated according to a weighted scheme including the various information resources and the current system context. Our integration algorithm works extremely robust. Moreover, it can easily be scaled up to additional input devices and various application domains.

1. INTRODUCTION

The development of user interfaces has become a significant factor in the software design process. Growing functional complexity and mostly restriction to purely haptic interaction required extensive learning periods and adaptation by the user to a high degree, which significantly increased user frustration. To overcome these limitations, various interface types and interaction paradigms have been introduced in the course of time. Multimodal interfaces currently resemble the latest step in this development, as they can be worked with easily, effectively and, above all, intuitively [1].

1.1. Multimodal Integration

In multimodal systems, information is provided by various input devices either in parallel or within short periods of time. These pieces of information have to be combined (integrated) in a meaningful way to interprete the user intention and generate appropriate system reactions. Besides managing redundant and complementary information streams the main problem is to handle competing user input[2].

Various approaches have been discussed to process multimodal input signals (e.g. [3] and [4]), differing either in the specific method (rule-based, statistical, etc.) or in the level of integration (feature fusion, late semantic fusion). To profit from the advantages of the individual approaches, several hybrid architectures have been introduced that combine these methods with regard to the current application[1].

1.2. Application Domain

Our research work focuses on the design of a multimodal interface for navigating in VRML worlds [5]. Conventional haptic devices can freely be combined with special Virtual-Reality hardware, and, as a key feature, with natural speech, as well as dynamic head and hand gestures. For exchanging information between the individual modules of the system we developed an extended context-free grammar formalism. As the grammar completely describes the functionality vocabulary of the application, it facilitates the representation of domain- and device independent multimodal information contents. Thereby, the terminal symbols of the grammar represent the smallest significant semantic units (called *semuns*) of potential user interactions. Taking into account the current system context, the individual semuns are combined in a semantic unification process.

2. GENETIC MULTIMODAL INTEGRATION

Genetic algorithms (GAs) and other techniques related to evolutionary computation are adaptive statistical methods, well established to solve complex optimization problems[6]. Based on the principles of evolution, they work in direct analogy of natural behavior. A population of *individuals*, each representing a solution for a problem, compete with each other. Characteristics of the best solutions are combined to produce new solutions. Randomly some characteristics are mutated, introducing new characteristics. Although GAs do not guarantee to find the global optimum solution, they have proved to find good solutions very quickly.

From: ICASSP 2002, IEEE Signal Processing Society

2.1. System overview

The structural elements of our genetic algorithm for integrating multimodal user input are shown in figure 1. A new run of the integration process is initiated when any of the connected devices reports a new user input. At first, an initial population is produced. Each of the individuals is rated according to its *fitness*, including an estimation of the certainty for the individual and the appropriate system command. In the next step, some of the individuals are chosen for recombination. The selected individuals are recombined, using two problem adapted genetic operators. For the new individuals, the command, its certainty, as well as a confidence measure is calculated. The fitness of all individuals has to be recalculated, since the population fitness has changed, and the fitness of an individual depends on the fitness of the whole population. Afterwards, the next generation is selected. The best individuals are chosen from the parents and childs and inserted in the next generation. If the population has converged, an appropriate system command can be generated. Otherwise the algorithm is iterated. In the case of any new input, additional data is added to the individuals. Thus, new recognition results and devices can be included online in the integration process.

2.2. Problem adapted natural coding

In a GA, every individual in a population represents a solution for a given problem, consisting of various parameters (*genes*). Each gene describes one parameter. Parameters belonging together are joined to form a set of values (*chromosomes*). All chromosomes represent one *genotype*. This genotype contains the complete information to generate a solution, i.e. a concrete system command (*phenotype*). The decoding function maps the genotype on the phenotype.

Following [7], our GA uses an abstract data type as non-standard natural coding that is specially adapted to the integration problem. Moreover, to allow meaningful recombination, application specific knowledge is incorporated in the genetic operators. A potential solution consists of three chromosome parts. The *administration* chromosome contains information about the pre-command (supplied with exact time and certainty), alternative pre-commands of previous integration steps, mutation possibilities and potential following commands as well as matching partners for complementary interactions. The *command* chromosome contains the generated system commands, i.e. a word of the grammar formalism with a specific certainty and a confidence measure, as well as information about absolutely necessary additional commands. Finally, a number of *recognizer* chromosomes, one for each input device, compromise detailed information on the recognized semantic units, recognition times, certainties, as well as lists of complementary, supplementary and competing information contents.

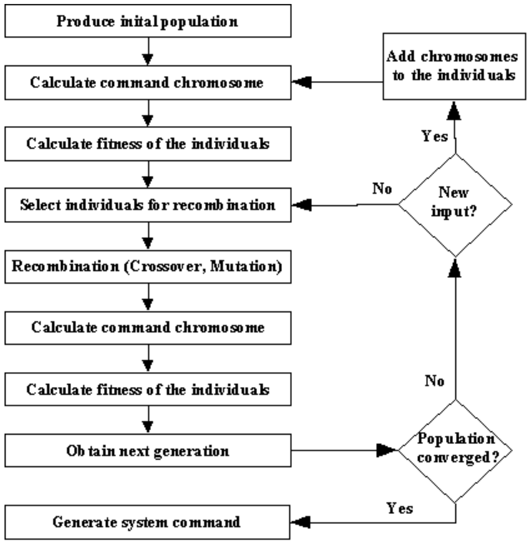
Produce inital population
Add chromosomes to the individuals
Calculate command chromosome
Yes
Calculate fitness of the individuals
No
New input?
Select individuals for recombination
Recombination (Crossover, Mutation)
Calculate command chromosome
No
Calculate fitness of the individuals
Obtain next generation
Population converged?
Yes
Generate system command

Fig. 1. Structural elements of the genetic algorithm

2.3. Creating an initial population

At first, the preset number of individuals for the population is created. Each individual is initialized with one administration and one command chromosome. While the command chromosome is kept empty, the data of the administration chromosome is copied from the last generated command, being equal for all individuals in the start population. For each input device, a *standard* recognizer chromosome is inserted, even if specific device is not currently active. For example in a bimodal system with speech and keyboard input, two standard recognizer chromosomes are inserted in the phenotype. If a devices reports more than one recognized semun, *extra* chromosomes are added to the phenotype. Finally, the generated chromosomes are filled with data either from the input devices or from a look-up-table (LUT), as far as general information is needed.

2.4. Determine the command chromosome

The command chromosome represents a potential integration result. Apart from the trivial case that all filled recognizer chromosomes contain the same semun, special rules have to be applied to estimate the consistence of the solution. If a specific semun dominates the phenotype, it is chosen, possibly in combination with a dominating complementary semun. If no single semun dominates the phenotype, the algorithm searches for potential complementary semuns. In the case of multiple complementary semuns, either the combination of a dominating semun and a dominating complementary semun or the combination with the highest overall certainty is selected.

2.5. Measuring the quality of a solution

The quality of a command chromosome is calculated due to a statistically weighted sum of the chromosomes in the genotype. To rate a semun with regard to its occurrence, a time factor is integrated in the calculation process. The final certainty is a probability that has to be interpreted carefully with regard to the number of connected devices.

The first step in the calculation process is to multiply the certainties of the recognizer chromosomes with the general certainty of the specific input device obtained from a LUT. This results in probabilities that depend on both the individual recognition results and the type of recognition modules.

In the second step, the certainty is calculated as follows: if N is the total number of recognizer chromosomes in the phenotype, F is the number of chromosomes representing a specific command c_g of the grammar, and p_i represents the modified probability of chromosome i, the overall certainty P is calculated by $P(c_g) = (\sum_{i=1}^{F} p_i)/N$. All certainties representing c_g are summarized and standardized.

In the next step, chromosomes that contain competing information, are time weighted and subtracted from the overall certainty. Concerning complementary information in the phenotype, the certainty has to be increased. Thus, chromosomes containing complementary semuns are time scaled and added to the overall certainty. Furthermore, information about the precommand is integrated in the calculation. If the generated command is complementary to the precommand, its certainty is time weighted and added to the overall probability, otherwise, if the precommand and the generated command do not evaluate to a correct word of the grammar, its certainty is subtracted.

Finally, the generated command is checked for completeness. If the command already represents a correct word, its certainty is left unchanged. In the other case, the certainty of the generated command is time scaled itself. Thus even an incomplete command can still reach a high certainty. The probability of a following complementary semun is high, but, concerning an old recognition result, that probability is very low. Therefore, the overall certainty is getting lower with time, if the command is not completed.

Merging the individual factors, the certainty of the command $P(c_g)$ can be calculated: Assume N, F, c_g and i as introduced above. Let G denote the number of recognizer chromosomes representing competing information for c_g and tp_j is the time weighted certainty of chromosome j. Furthermore assume that H describes the number of recognizer chromosomes that represent complementary information to c_g and tp_k is the time scaled certainty of chromosome k. Finally, let tp_p denote the certainty of the precommand (incompatible semuns modeled by negative values).

$$P(c_g) = \frac{\sum_{i=1}^{F} p_i - \sum_{j=1}^{G} tp_j + \sum_{k=1}^{H} tp_k}{N} + tp_p$$

Calculating the fitness of the individuals in a population is then just a straight forward process, because the certainty data already provides a measure of the quality of a phenotype. This measure simply needs to be standardized and compared to the other individuals. Therefore, the overall certainties in the population are summed, giving a measure for the fitness of the whole population. The individual fitness of an individual resolves to its certainty divided by the fitness of the whole population.

2.6. Selection process

In the parent selection process, individuals are chosen for recombination and producing offspring. The genes of these individuals are combined according to genetic crossover and randomly changed by mutation. Fitter individuals are more likely to be selected for the recombination process than weak members of the population. In *standard* selection, individuals are chosen in direct proportion to their fitness. As a disadvantage of this method, the existence of super fit individuals can disturb the whole evolution process. Therefore, *tournament* selection is preferred. Individuals of the population compete against each other in a tournament with a predefined tournament size T. T individuals from the population are randomly chosen, independently from their fitness. The individual in the tournament with the highest fitness is selected in the mating pool. As an alternative option, not only the winner, but the second and the third placed individuals may by chosen, too. The selection pressure is proportional to the tournament size T.

2.7. Recombination process

The two standard operators crossover and mutation have been adapted to multimodal integration. While crossover is the dominating technique for a fast exploration of the entire search space, mutation guarantees, that no possibility is assigned to the probability zero. Moreover, mutation helps to avoid convergence on local maxima.

In the recombination phase, the selected individuals are first combined with crossover to produce new offspring. Information saved in the administration and the recognizer chromosomes may be transposed between the individuals. Offspring inherits randomly chosen chromosomes from its parents. The more input devices are connected to the system, it is more likely that the recognizer chromosomes are well mixed from the parents, leading to different solutions.

Some individuals of the new population, containing both parents and childs, are then changed by the mutation operator. For this purpose, in a predefined LUT for each semun, mutation possibilities, and probabilities are stored that are evaluated to change the administration and the recognition chromosomes. The command chromosome is not changed, but recalculated each time a new individual is created.

2.8. Convergence of the population

For estimating the convergence of a population, two fundamentally different criteria can be specified. The first approach accepts a solution, if a predefined value for the population fitness is reached. Since the absolute fitness values of the individuals depend on the number of connected input devices and the device types, they may strongly change from run to run. The second approach is more appropriate for multimodal integration. It accepts a solution, if nearly all individuals share the same genes. If a predefined number of individuals has reached the same fitness as the best individual in the population, the population has converged.

3. PROTOTYPICAL EVALUATION

The multimodal integration algorithm has been implemented in a prototypical environment and evaluated with regard to various factors. The population size should be kept constant as differing sizes introduce various scaling problems, and, in general, lead to worse recognition results. If the start population is initialized with the *n*-best lists of the connected input devices instead of choosing the same value for each individual, the convergence performance is improved.

An exponential time scaling function normally results in improved recognition results. But if several not necessarily realtime capable recognition modules are connected (e.g. a natural speech module) to the interface, a linear time weight function reduces the potential of unintentional devaluation of the appropriate multimodal information.

In the recombination process, we found out that the individuals have to be provided with more extra chromosomes than actually necessary. Therefore, the crossover operator can be extended, leading to a significantly enlarged search space and more accurate integration results. Mutation rates should be higher than in classical GAs (typically 0.01 - 0.05) and employed with regard to the probabilities of the individual semuns. Moreover, it is important that mutation is only allowed for the identified possibilities. Otherwise, the algorithm often drifts to statistically good solutions that do not represent correct command sequences and thus requires the introduction of a repair function.

4. FUTURE WORK

For the nearest future, we further plan to improve the integration algorithm. Especially the calculation of the command chromosome offers various improvements. If some semuns dominate the phenotype, but none of them has a dominating complementary chromosome, the generated command gene may be chosen randomly from the dominating semuns. This enlarges the search space, but ignores recognition probabilities from the input devices. As a second possibility, if some semuns dominate the phenotype, but none of them has a dominating complementary chromosome, new individuals may be created. For each dominating complementary chromosome, a new individual is created. The population size is not constant, but enlarges with this operator.

Due to genetic drift, GAs normally do not stay on local maxima. If *n*-best lists for the integration results are to be produced, the next generation selection is to be modified: Child and parent generation do not compete any longer against each other in a tournament selection. Each child is just compared with its parents. If the child has similar attributes compared to one of its ancestors and at the same time a higher fitness, the ancestor is replaced by its child. Together with a new convergence criteria, similar attributes are kept constant during evolution and the algorithm may find not only the global optimum, but also local maxima and can produce a *n*-best list of commands. As a next step the identified options are to be evaluated and compared in different multimodal systems. Finally, the influence of detailed user data from usability studies is to be evaluated.

5. CONCLUSIONS

In this paper, a new approach towards combining multimodal user input has been presented. The integration technique is based on evolutionary computation: different integration results compete against each other and are recombined by problem adapted crossover and mutation operators. Handling redundant, complementary, and competing information is implicitly modeled by the algorithm. The integration method can easily be scaled up to additional input devices and different application domains.

6. REFERENCES

[1] S. L. Oviatt, “Multimodal interface research: A science without borders,” *Proc. of the 6th Int. Conf. on Spoken Language Processing (ICSLP)*, Beijing, China 2000.

[2] L. Nigay et al., “A generic platform for adressing the multimodal challenge,” in *Proc. of CHI ’95*, 1995.

[3] A. Waibel et al., “Multimodal interfaces,” *Artifical Intelligence Review*, vol. 10, no. 3-4, pp. 299–319, 1996.

[4] L. Wu et al., “Multimodal integration - a statistical view,” *IEEE Trans. on Multimedia*, vol. 1, Dec. 1999.

[5] F. Althoff et al., “Using multimodal interaction to navigate in arbitrary virtual worlds,” *In WS on Perceptive User Interfaces (PUI 01)*, Nov., Orlando, USA 2001.

[6] D. Fogel, *Evolutionary Computation*, IEEE Press, ’95.

[7] Z. Michalewicz, *Genetic Algoithms and Data Structures*, Springer-Verlag, New York, 1999.