27

Virtual Reality and Computational Design

Michael S. Bittermann and I. Sevil Sariyildiz
Delft University of Technology
The Netherlands

1. Introduction

Virtual reality (VR) has been used for diverse purposes, including medical surgery training, visualizing metabolic pathways, socio psychological experiments, flight and driving simulation, as well as industrial and architectural design [1-5]. In these applications the role of VR is to represent objects for visual experience by a human expert. In engineering and design applications the purpose is to verify the performance of a design object with respect to the criteria involved in the task during a search for superior solutions. In computational design, where this verification and search process are performed by means of computation, the instantiation of objects in virtual reality may become a necessary feature. The necessity occurs when the verification process requires the presence of 'physical' object attributes beyond the parameters that are subject to identification through search. For example, in an architectural design the goal may be to determine the most suitable position of an object, while the suitability is verified based on visual perception characteristics of the object. That is, the verification requires the presence of object features beyond the object's location in order to exercise the evaluation of the object's performance regarding the perception-based requirements. These features are provided when the object is instantiated in VR. This way a measurement process driving the evaluation, such as a virtual perception process in the form of a stochastic sampling process, can be executed to assess the perceptual properties of the object concerned.

This paper elucidates the role of VR in computational design by means of two applications, where VR is a necessary feature for the effectiveness of the applications. The applications concern a computational design system implemented in VR that identifies suitable solutions to design problems. The effectiveness of the system has been established in previous work [6], while the general significance of the role that VR plays in the system has not been addressed. This will be accomplished in this paper, which is organized as follows. In section two the computational system is described. In section three the role that VR is playing in the system is described and demonstrated with two applications from the domain of architectural design. This is followed by conclusions.

2. A computational design system implemented in Virtual Reality

In several instances during a design process VR enables decision makers to better comprehend the implications of design decisions. Two aspects can be distinguished in this process.

First, a design's implication in terms of the degree that it satisfies the objectives pursued is subject to assessment. This process may be termed as *verification,* as it entails the verification of the requirements' satisfaction during a search to maximize the satisfaction degree. It is noted that the concept of Pareto optimality plays an important role in the search for optimality. Namely in general it is problematic for a decision maker to commit himself for a specific relative importance among the major goals for the design at hand prior to knowing the implications of such a commitment. This is due to the generally abstract nature of the goals in design. For example the aims to have high functionality or low cost, clearly are difficult to put in perspective prior to knowing what solutions may be attained when maximizing the satisfaction of these goals in the present task. Pareto optimality addresses this issue by permitting to postpone the commitment on relative importance until a set of equivalent solutions is obtained that cannot be improved further. This is achieved by establishing those solutions where no others exist that outperform them in all goals at the same time.

A second process concerns *validation* of the objectives. That is, the question if the right objectives are pursued during verification is addressed. The latter process requires insights beyond knowing how to reach optimality for the given goals at hand. Namely contingent requirements that have not been put into the play during verification are to be pin-pointed. It is clear that the latter process requires verification to occur before it, since otherwise there is no rationale to modify the objectives. That is, based on the Pareto optimal solutions found, a designer is to compare these solutions against his/ her preferences, yielding clues on the modification of criteria. The relation between the verification and validation process in design are shown in figure 1.

The reason why VR facilitates validation is that it allows considering the solution in the physical domain beyond an abstract description of the targeted performance features, so that a decision maker may become aware of directions for modifying the objectives. The validation process is especially soft, since it is highly contingent to circumstances so that potentially a vast amount of desirable objectives may be subject to inclusion in a design task, and it is generally problematic to have a hint about which ones to include as well as their relative importance [7]. Therefore it is a challenging issue to provide computational support for the validation process.

In order to investigate the role of VR in the search for optimality during verification, we take a closer look at verification and its associated search process. A computational system accomplishing this task is shown in figure 2. It aims to establish set of Pareto optimal solutions for a number of requirements, where the requirements are allowed to be *soft* in character, i.e. they may contain imprecision and uncertainty.

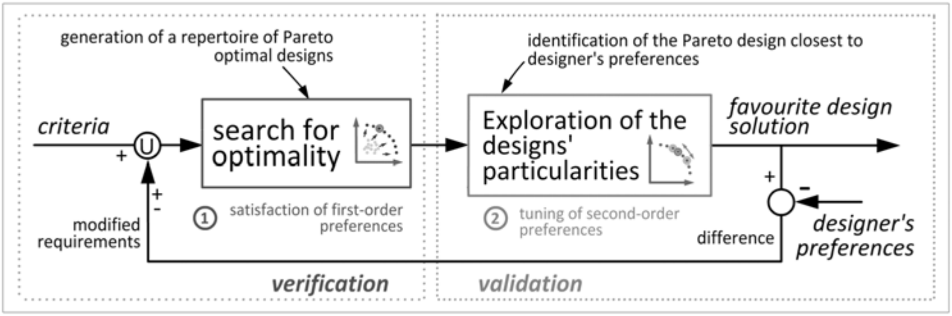

generation of a repertoire of Pareto optimal designs
identification of the Pareto design closest to designer's preferences
criteria
+
U
search for optimality
Exploration of the designs' particularities
favourite design solution
+
-
+
-
modified requirements
① satisfaction of first-order preferences
② tuning of second-order preferences
difference
designer's preferences
verification
validation

Fig. 1. Verification and validation in design

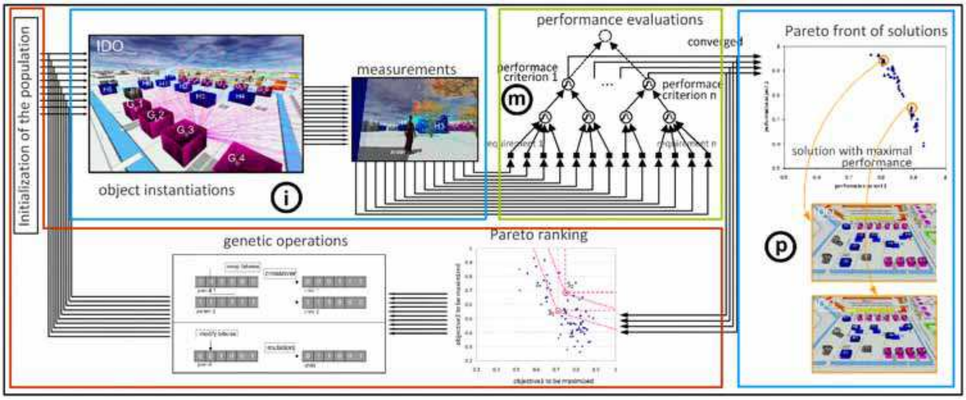


Fig. 2. Computational design system implemented in virtual reality

From the figure it is noted that the system consists of four components: a multi-objective genetic algorithm; a neuro-fuzzy model; object instantiation in VR; and instantiation of Pareto optimal solutions in VR. The genetic algorithm is marked by the red box, the fuzzy model is marked by the green box. The two components involving VR are shown in the blue boxes. In order to pin-point the role VR plays in the system, first it is necessary to explain the evolutionary and the fuzzy system components. The role of VR is described in section three.

2.1 Evolutionary search for multi-objective optimality

The task of the multi-objective search algorithm in the design system above is to gear the process towards desirable solutions. Multi-objective optimization deals with optimization where several objectives are involved. In design generally multiple objectives are subject to simultaneous satisfaction. Such objectives e.g. are high functionality and low cost. These objectives are conflicting or in competition among themselves. For a single objective case there are traditionally many algorithms in continuous search space, where gradient-based algorithms are most suitable in many instances. In discrete search spaces, in the last decade evolutionary algorithms are ubiquitously used for optimization, where genetic algorithms (GA) are predominantly applied. However, in many real engineering or design problems, more than two objectives need to be optimized simultaneously. To deal with multi-objectivity, evolutionary algorithms with genetic operators are effective in defining the search direction for rapid and effective convergence [8]. Basically, in a multi-objective case the search direction is not one but may be many, so that during the search a single preferred direction cannot be identified and even this is not desirable. To deal with multi-objectivity evolutionary algorithms are effective in defining the search direction, since they are based on a population of solutions. Basically, in a multi-objective case the search direction is not one but may be many, so that during the search a single preferred direction cannot be identified. In this case a population of candidate solutions can easily hint about the desired directions of the search and let the candidate solutions during the search process be more probable for the ultimate goal. Essential machinery of evolutionary algorithms is the principles of GA optimization, which are the genetic operations. Genetic operations entail the probabilistic combination among favourable solutions in order to provoke the emergence of more suitable solutions. Use of these principles is inspired from the phenomenon of biological evolution. It proves to be effective for multi-modal objective functions, i.e. problems involving many local optima. Therefore the evolutionary approach is robust and suitable for real-world problems.

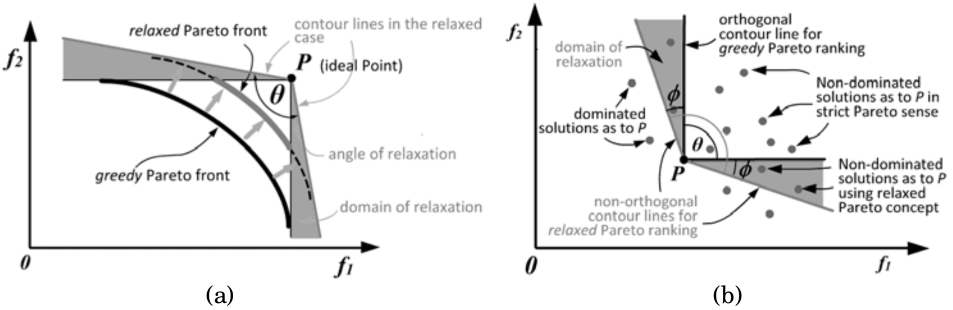
Next to the evolutionary principles, in Multi-objective (MO) algorithms, in many cases the use of Pareto ranking is a fundamental selection method. Its effectiveness is clearly demonstrated for a moderate number of objectives, which are subject to optimization simultaneously. Pareto ranking refers to a solution surface in a multidimensional solution space formed by multiple criteria representing the objectives. On this surface, the solutions are diverse but they are assumed to be equivalently valid. The driving mechanism of the Pareto ranking based algorithms is the conflicting nature of criteria, i.e. increased satisfaction of one criterion implies loss with respect to satisfaction of another criterion. Therefore the formation of Pareto front is based on objective functions of the weighted N objectives $f_1, f_2,\ldots, f_N$ which are of the form

$$F_i(\boldsymbol{x}) = f_i(\boldsymbol{x}) + \sum_{j=1, j\neq i}^{j=N} a_{ji} f_j(\boldsymbol{x}), i = 1,2,...,N \tag{1}$$

where $F_i(x)$ are the new objective functions; a_{ji} is the designated amount of gain in the j-th objective function for a loss of one unit in the i-th objective function. Therefore the sign of a_{ji} is always negative. The above set of equations requires fixing the matrix a. This matrix has all ones in its diagonal elements. To find the Pareto front of a maximization problem we assume that a solution parameter vector $\boldsymbol{x}_1$ dominates another solution $\boldsymbol{x}_2$ if $F(\boldsymbol{x}_1)\geq F(\boldsymbol{x}_2)$ for all objectives. At the same time a contingent equality is not valid for at least one objective.

In solving multi-objective optimization, the effectiveness of Pareto-ranking based evolutionary algorithms has been well established. For this purpose there are quite a few algorithms which are running quite well especially with low dimensionality of the multidimensional objective space [9]. However, with the increase of the number of objective functions, i.e. with high dimensionality, the effectiveness of the evolutionary algorithms is hampered. Namely with many objectives most solutions of the population will be considered non-dominated, although the search process is still at a premature stage. This means the search has little information to distinguish among solutions, so that the selection pressure pushing the population into the desirable region is too low. This means the algorithm prematurely eliminates potential solutions from the population, exhausting the exploratory potential inherent to the population. As a result the search arrives at an inferior Pareto front, and with aggregation of solutions along this front [10]. One measure of effectiveness is the expansion of Pareto front where the solution diversity is a desired property. For this purpose, the search space is exhaustively examined with some methods, e.g. *niched Pareto ranking*, e.g. [11]. However these algorithms are rather involved so that the search needs extensive computer time for a satisfactory solution in terms of a Pareto front. Because of this extensive time requirement, distributed computing of Pareto-optimal solutions is proposed [12], where multiple processors are needed.

The issue of solution diversity and effective solution for multi-objective optimization problem described above can be understood considering that the conventional Pareto ranking implies a kind of *greedy* algorithm which considers the solutions at the search area delimited by orthogonal axes of the multidimensional space, i.e. a_{ji} in Eq. 1 becomes zero. This is shown in figure 3 by means of the orthogonal lines delimiting the dominated region.


f2
relaxed Pareto front
contour lines in the relaxed case
P (ideal Point)
θ
angle of relaxation
greedy Pareto front
domain of relaxation
0
f1
f2
domain of relaxation
orthogonal contour line for greedy Pareto ranking
Non-dominated solutions as to P in strict Pareto sense
dominated solutions as to P
ϕ
θ
P
ϕ
Non-dominated solutions as to P using relaxed Pareto concept
non-orthogonal contour lines for relaxed Pareto ranking
0
f1

(a) (b)

Fig. 3. Contour lines defining the dominated region in relaxed versus greedy case (a); implementation of the relaxation concept during the evolutionary search process (b)

The point P in figure 3a is ultimately subject to identification as an ideal solution. To increase the pressure pushing the Pareto surface towards to the maximally attainable solution point is the main problem, and relaxation of the orthogonality with a systematic approach is needed and applied in this work. From figure 3a it is noted that by increasing the angle at P from the conventional orthogonal angle to a larger angle implies that the conventional dominated region is expanded by domains of relaxation. This also entails that theoretically a Pareto front is to be reached that is located closer towards the ideal Point P.

Such an increase of the angle delimiting the search domain implies a deviation from the conventional concept of Pareto dominance, namely the strict Pareto dominance criterion is relaxed in the sense that next to non-dominated solutions also some dominated solutions are considered at each generation. This is seen from figure 3b, where the point $\boldsymbol{P}$ denotes one of the individuals among the population in the context of genetic algorithm (GA) based evolutionary search. In the greedy search many potential favourable solutions are prematurely excluded from the search process. This is because each solution in the population is represented by the point $\boldsymbol{P}$ and the dominance is measured in relation to the number of solutions falling into the *search domain* within the angle $\theta=\pi/2$. To avoid the premature elimination of the potential solutions, a relaxed dominance concept is implemented where the angle θ can be considered as the *angle for tolerance* provided $\theta>\pi/2$. The resulting Pareto front corresponds to a non-orthogonal *search domain* as shown in figure 3. The wider the angle beyond $\pi/2$ the more tolerant the search process and vice versa. For $\theta<\pi/2$, θ becomes the *angle for greediness*. Domains of relaxations are also indicated in Figure 3b. In the greedy case the solutions are expected to be more effective but to be aggregated. In the latter case, the solutions are expected to be more diversified but less effective. That is because such dominated solutions can be potentially favourable solutions in the present generation, so that they can give birth to non-dominated solutions in the following generation.

Although, some relaxation of the dominance is addressed in literature [13, 14], in a multidimensional space, to identify the size of relaxation corresponding to a volume is not explicitly determined. In such a volume next to non-dominated solutions, dominated but potentially favourable solutions, as described above, lie. To determine this volume optimally as to the circumstantial conditions of the search process is a major and a challenging task. The solution for this task is essentially due to the mathematical treatment of the problem where the volume in question is identified adaptively during the search that it yields a measured pressure to the Pareto front toward to the desired direction, at each generation as follows.

The fitness of the solutions can be ranked by the fitness function

$$\boldsymbol{R}_{fit} = \frac{1}{N(\theta)+n} \tag{2}$$

where n is the number of potential solutions falling into the *search domain* consisting of the conventional orthogonal quadrant, with the added areas of relaxation. To obtain n in Eq. 2, for each solution point, say P in Figure 3b, the point is temporarily considered to be a reference point as origin, and all the other solution points in the orthogonal coordinate system are converted to the non-orthogonal system coordinates. This is accomplished by means of the matrix operation given by Eq. 3 [15],

$$F = \begin{bmatrix} F_1 \\ F_2 \\ \cdots \\ F_n \end{bmatrix} = \begin{bmatrix} 1\ a_{21}\ \ldots\ldots\ a_{n1} \\ a_{12}\ 1\ \ldots\ldots\ a_{1n} \\ \ldots\ldots\ldots\ldots \\ a_{1n}\ a_{2n}\ \ldots\ldots\ 1 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ \ldots\ldots \\ f_n \end{bmatrix} = \begin{bmatrix} 1 \quad \tan(\phi_2)\ \ldots\ldots\ \tan(\phi_n) \\ \tan(\varphi_2) \quad 1\ \ldots\ldots \quad \tan(\varphi_n) \\ \ldots\ldots\ldots\ldots \\ \tan(\theta_2)\ \tan(\theta_n)\ \ldots\ldots \quad 1 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ \ldots\ldots \\ f_n \end{bmatrix} \tag{3}$$

where the angles ϕ, φ, … θ represent the respective relaxation angles between one axis of the coordinate system and the other axes. After coordinate transformation using Eq 3, all points which have positive coordinates in the non-orthogonal system correspond to potential solutions contributing to the next generation in the evolutionary computation. If any point possesses a negative component in the new coordinate system, the respective solution does not dominate P and therefore is not counted. This is because otherwise such a solution may lead the search in a direction away from P. The importance of this coordinate transformation becomes dramatic especially with greater amounts of objective dimensions. In such cases the spatial distribution of domains of relaxation becomes complex and is therefore difficult to implement. Namely, in multidimensional space the volume of a relaxation domain is difficult to imagine, and more importantly it is difficult to identify the population in such domains. Therefore many different methods for effective Pareto front formation in the literature [10, 16] are reported. However Eq. 3 provides a decisive and easy technique revealed in this work for the same goal. The approach through the coordinate transformation is a systematic and elegant approach, alleviating the bottleneck of conventional Pareto ranking dealing with many objectives to some extend, so that the evolutionary paradigm becomes more apt for applications in design usually containing a great many requirements.

In order to maximize the effectiveness of the relaxation, the determination of the suitable relaxation angle is a contingent issue, i.e. it depends on the particular conditions occurring during the stochastic search process. For instance, during a prematurely developed Pareto front, applying large relaxation angles may not permit effective distinction among the solutions regarding their suitability for the ultimate goal. Or during later stages of Pareto front development, a smaller angle will exhaust the diversity in the population and thus diminish the selection pressure towards the desirable regions. This means fixing the relaxation angle in advance may not be able to let the population arrive at a Pareto front as close to the ideal point compared to a strategy where the angle is adaptively changing during the search, taking the present conditionality of the Pareto front into account.

Adaptively changing the angle implies that the angle used to grade the individual solution's suitability is considered in perspective with the relaxation angles presently associated to the other solutions in the population. This is implemented by means of Eq. 4, where the ratio between the relaxation angle and average relaxation angle is used. *N(θ)* in Eq. 4 can be considered as expressing the amount of virtual solutions that are accrued to the counted number of dominant solutions given by *n* in Eq. 2, reflecting the fact that when we take the greedy dominance concept solutions that are dominated by *s* more solutions may turn out to be favourable in the search process although they normally would be eliminated due to greediness of the algorithm.

$$N(\theta) = \frac{s}{1 + \left(\theta / \overline{\theta}\right)} \tag{4}$$

Considering Eq. 2 and Eq. 4 together it is clear that the purpose is to reward a chromosome for affording a wide relaxation angle *θ*, relative to the average angle of the population $\overline{\theta}$, and still having a low dominance count, denoted by *n*. The wide angle provides more diversity in the population for the next generation. However, when the relaxation angle would be excessively big, the population for the next generation can be crowded with trivial solutions. To prevent that, in Eq. 2 the number of non-dominated solutions with respect to the particular solution considered denoted by *n*, is summed up with the function of the angle *N(θ)*. This means that between two solutions with the same amount of non-dominated solutions, the one with the wider angle is preferred. This is done for every solution in the population. This implies that the average angle $\overline{\theta}$ is changing for every generation adaptively. It is noted that the number *s* appearing in Eq. 4 is a constant number, used to adjust the relative significance of relaxation angle versus count *n*. This means the value of *s* should be selected bearing in particular the population size in mind, so that for instance solutions using wide angles are adequately rewarded.

2.2 A fuzzy model for performance evaluation

The fuzzy model marked by the letter *m* in figure 2 enables the multi-objective search process to evaluate the solutions it generates and combines genetically, using some human-like reasoning capabilities. That is, the solutions are evaluated with respect to complex, vague objectives having a linguistic character. Design tasks, in particular in the domain of built environment, involve goals with such properties, e.g. functionality, or sustainability. During the search for optimality in design the suitability of a solution for the goals needs to be estimated. This means beyond observing the direct physical features of a solution, they need to be interpreted with respect to the goals pursued. For example, designing a space it may be desirable that the space is *large* or it is *nearby* another space. Clearly these requirements have to do with the size of the space, and the distance among spaces respectively, which are physical properties of the design. However, it is clearly noted that largeness is a concept, i.e. it does not correspond immediately to a physical measurement, but it is an abstract feature of an object. It is also noted that there is generally no sharp boundary from on which one may attribute such a linguistic feature to an object. For instance there is generally no specific size of a room from on which it is to be considered large, and below which it is not large. Many design requirements have this character, i.e. they do not pin-point a single acceptable parameter value for a solution, but a range of

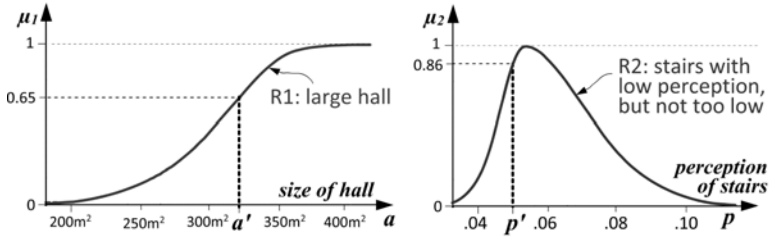


Fig. 4. Two fuzzy sets expressing two elemental design requirements

values that are more or less satisfactory. This is essentially because design involves conflicting requirements, such as spaciousness versus low cost. Therefore many requirements are bound to be merely partially fulfilled. Such requirements characterized as *soft*, and they can be modelled using fuzzy sets and fuzzy logic from the soft computing paradigm [17]. A fuzzy set is characterized via a function *termed fuzzy membership function* (mf), which is an expression of some domain knowledge. Through a fuzzy set an object is associated to the set by means of a membership degree μ. Two examples of fuzzy sets are shown in figure 4. By means of fuzzy membership functions a physical property of a design, such as size, can be interpreted as a degree of satisfaction of an elemental requirement. The degree of satisfaction is represented by the membership degree.

The requirements considered in figure 4 are relatively simple, whereas the ultimate requirement for a design - namely a high design performance - is complex and abstract. Namely the latter one is determined by the simultaneous satisfaction of a number of elemental requirements.

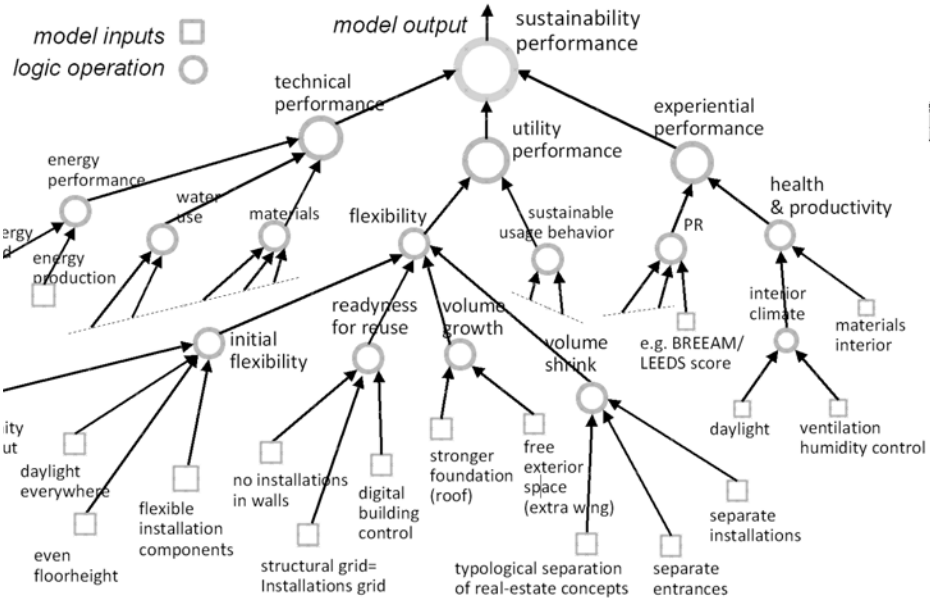


Fig. 5. The structure of a fuzzy neural tree model for performance evaluation

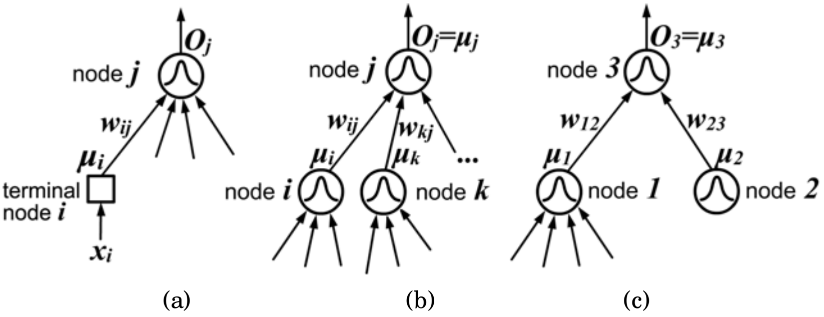


Fig. 6. Different type of node connections in the neuro-fuzzy model in figure 5

In this work the performance is computed using a fuzzy neural tree. It is particularly suitable to deal with the complex linguistic concepts like performance of a design. A neural tree is composed of one or several model output units, referred to as *root nodes* that are connected to input units termed *terminal nodes*, and the connections are via logic processors termed *internal nodes.* An example of a fuzzy neural tree for performance evaluation of a design is shown in Figure 5. The neural tree is used for the evaluation by structuring the relations among the aspects of performance. The root node takes the meaning of *high sustainability performance* and the inner nodes one level below are the aspects of the performance. The meaning of each of these aspects may vary from design project to project and it is determined by experts. The model inputs are shown by means of squares in Figures 5 and 6, and they are fuzzy sets, such as those given in Figure 4.

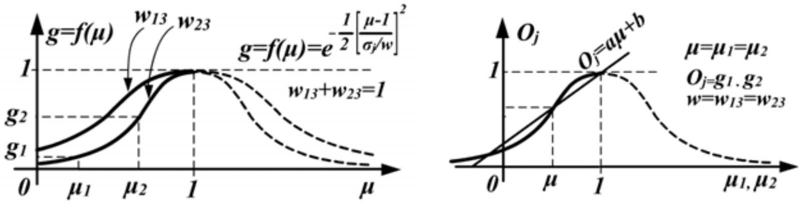
The detailed structure of the nodal connections with respect to the different connection types is shown in Figure 7, where the output of i-th node is denoted μ_i and it is introduced to another node j. The weights w_{ij} are given by domain experts, expressing the relative significance of the node i as a component of node j.

The centres of the basis functions are set to be the same as the weights of the connections arriving at that node. Therefore, for a *terminal node* connected to an *inner node*, the inner node output denoted by O_j, is obtained by [18].

$$O_j = \exp(-\frac{1}{2}\sum_i^n \left[\frac{(\mu_i - 1)}{\sigma_j / w_{ij}}\right]^2) \tag{5}$$

where j is the number of the node; i denotes consecutive numbers associated to each input of the inner node; n denotes the highest number of the inputs arriving at node j; w_i denotes the degree of membership being the output of the *i-th* terminal node; w_{ij} is the weight associated with the connection between the i-th terminal node and the inner node j; and σ_j denotes the width of the Gaussian of node j.

It is noted that the inputs to an inner node are *fuzzified* before the AND operation takes place. This is shown in Figure 7a. It is also noted that the model requires establishing the width parameter σ_j at every node. This is accomplished by means of imposing a consistency condition on the model [18]. This is illustrated in figure 7b where the left part of the Gaussian is approximated by a straight line. In figure 7b, optimizing the σ_j parameter, we obtain



(a) (b)

Fig. 7. Fuzzification of an input at an inner node (a); linear approximation to Gaussian function at AND operation (b)

$$O_j \cong \mu \tag{6}$$

for the values μ and O_j can take between zero and one. In any case, for a node in the neural tree, Eq. 6 is satisfied for $\mu=O_j=0$ (approximately) and for $\mu=O_j=1$ (exact) inherently, while g_1 and g_2 are increasing function of μ_1 and μ_2. Therefore a linear relationship between O_j and μ in the range between 0 and 1 is a first choice from the fuzzy logic viewpoint; namely, as to the AND operation at the respective node, if inputs are equal, that is $\mu=\mu_1=\mu_2$ then the output of the node of μ_1 AND μ_2 is determined by the respective *triangular membership functions* in the antecedent space. Triangular fuzzy membership functions are the most prominent type of membership functions in fuzzy logic applications. For five inputs to a neural tree node, these membership functions are represented by the data sets given by Table 1 and Table 2.

.1	.2	.3	.4	.5	.6	.7	.8	.9
.1	.2	.3	.4	.5	.6	.7	.8	.9
.1	.2	.3	.4	.5	.6	.7	.8	.9
.1	.2	.3	.4	.5	.6	.7	.8	.9
.1	.2	.3	.4	.5	.6	.7	.8	.9

Table 1. Dataset at neural tree node input

.1	.2	.3	.4	.5	.6	.7	.8	.9

Table 2. Dataset at neural tree node output

In general, the data sets given in Table 1 and Table 2 are named in this work as '*consistency conditions*'. They are used to calibrate the membership function parameter σ. This is accomplished through optimization. The consistency condition is to ensure that when all inputs take a certain value, then the model output yields this very same value, i.e. $\mu_1=\mu_2\approx O_j$ This is illustrated in Figure 7b by means of linear approximation to the Gaussian. The consistency is ensured by means of gradient adaptive optimization, identifying optimal σ_j values for each node. It is emphasized that the fuzzy logic operation performed at each node is an AND operation among the input components $\boldsymbol{\mu_i}$ coming to the node. This entails for instance that in case all elemental requirements are highly fulfilled, then the design performance is high as well. In the same way, for any other pattern of satisfaction on the elemental level, the performance is computed and obtained at the root node output. The fuzzy neural tree can be seen as a means to aggregate elemental requirements yielding fewer requirement items at higher levels of generalization compared to the lower level requirements. This is seen from Figure 8.

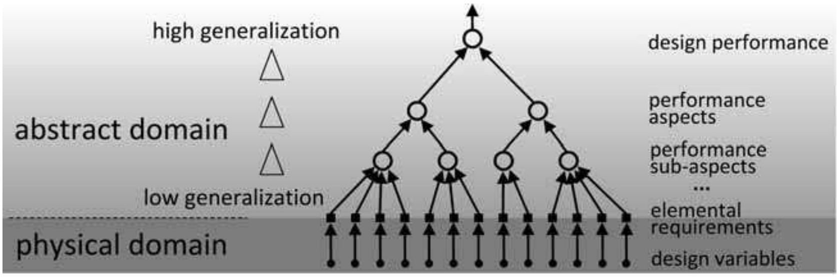


Fig. 8. Degrees of generalization in the neuro-fuzzy performance evaluation

At this point a few observations are due, as follows. If a weight w_{ij} is zero, this means the significance of the input is zero, consequently the associated input has no effect on the node output and thus also the system output. Conversely, if a w_{ij} is close to unity, this means the significance of the input is highest among the competitive weights directed to the same node. This means the value of the associated input is extremely important and a small change about this value has big impact on the node output O_j. If a weight w_{ij} is somewhere between zero and one, then the associated input value has some possible effect on the node output determined by the respective AND operation via Eq. 5. In this way, the domain knowledge is integrated into the logic operations. The general properties of the present neural tree structure are as follows: If an input of a node is small (i.e., close to zero) and the weight w_{ij} is high, then, the output of the node is also small complying with the AND operation; If a weight w_{ij} is low the associated input cannot have significant effect on the node output. This means, quite naturally, such inputs can be ignored; If all input values coming to a node are high (i.e., close to unity), the output of the node is also high complying with the AND operation; If a weight w_{ij} is high the associated input x_i can have significant effect on the node output. It might be of value to point out that, the AND operation in a neural-tree node is executed in fuzzy logic terms and the associated connection weights play an important role on the effectiveness of this operation.

3. The role of VR in the system

3.1 General considerations

From the descriptions of the two components in the previous section, it is clear that in order for the genetic algorithm to be effective, the suitability of the solutions it generates needs to evaluated using the fuzzy model. In conventional applications of Multi-objective GA, for instance maximizing the strength of a structural component and minimizing its weight at the same time, this evaluation is rather simple. The simplicity is in the sense that the fitness function is crisp and the parameters of the function, such as geometric parameters of the beam’s cross-section, are directly those parameters that are subject to evolutionary identification. In these cases there is no necessity for instantiation of the beam object during the search for optimality. However, in other search tasks, as they occur for instance in architectural design, the problem requires more elaborate treatment, in particular object instantiation in virtual reality. This necessity arises when the parameters that are subject to identification through the genetic algorithm cannot be used as parameters in a fitness function because the evaluation of fitness requires the information from other object features. As an example let us consider a problem, where optimal positions for a number of design elements are pursued, while the determination of the suitability of the positioning requires information on the perceptual properties of the objects. A virtual perception process is needed that obtains the required input information used in the human-like reasoning during the evaluation process. Obtaining the input information requires the instantiation of object features beyond the parameters that are subject to identification through the search.

This is seen from figure 2, where the role of VR in the design system is to permit instantiation of the candidate solutions, as indicated by the letter *i*, so that measurements required for the fuzzy performance evaluation are executed for these solutions. The measurements deliver input information for the human-like reasoning about the suitability of a solution using the neuro-fuzzy model marked with the letter *m*. With this understanding the role of VR in the search process can be considered as the *interface* between the two components evolutionary algorithm and fuzzy performance evaluation. In particular, referring to figure 8, the instantiation of objects in VR permits the execution of measurement procedures that deliver input information from the parameter domain for the interpretations with respect to the abstract goals.

It is noted that for the effective multi-objective optimization in the application below the relaxation angle is computed adaptively for every chromosome, and at every generation. This is implemented by having the angle be a part of the chromosome of every solution. The fitness of a chromosome is obtained by considering two properties of the solution at the same time. One is the degree of dominance in terms of the amount of solutions dominating an individual, the second is the relaxation angle used to measure this amount. Based on Eqs. 2 and 4 the fitness in the applications is assessed with s=20, i.e. explicitly

$$R_{fit} = \frac{1}{\dfrac{20}{1+\left(\theta / \overline{\theta}\right)} + n} \tag{7}$$

It is noted that the amount of chromosomes used in the tasks to be described in the following sections is 80.

Next to the need for object instantiation in VR during the search for optimal solutions there is a second instance during a design process when virtual reality plays a significant role. This is indicated by the letter *p* in figure 2, and concerns the investigation of the Pareto optimal solutions previously obtained. It is noted that generally multi-objective optimization involves no information on the relative importance among the objectives. This is in particular due to the abstract nature of the major goals making such a-priori commitment problematic. It is emphasized that in the present work this is the reason why the optimization takes place for the nodes at the penultimate neural level and not for the root node. Due to the lacking information on the relative importance among the criteria, generally Pareto optimal solutions cannot be distinguished without bringing into play higher-order criteria. That is, once a Pareto front is established, the difference among the solutions is subject to analysis, in order to determine a preference vector grading the objectives w.r.t. each other. In order for this process to be informative, it is required that the solutions found through evolutionary search be located at diverse positions on the Pareto front. This is to avoid that potentially interesting regions in objective space remain unexplored during the analysis of the Pareto front. It is emphasized that this diversity is obtained through the relaxation of the Pareto angle in this work.

With a diversely populated Pareto front it is possible to explore the front in a way that allows a decision-maker to intuitively grasp the relation between parameters of the solutions and corresponding performance characteristics, and in this way a decision maker is able to approach his most preferred solution among the Pareto optimal ones. Namely, the very nature of Pareto front implies that the trade-off that is afforded when moving along the Pareto surface is the inevitable trade-off inherent in the problem. This means, in case one is moving along the Pareto surface in a certain direction, for example towards better cost performance, the reduction of performance in the other dimensions, say the loss of functionality, is as small as possible through the definition of Pareto front. This means when a decision maker is observing a solution instantiated in virtual reality, i.e. in the parameter domain, he may decide to move in objective domain into the direction he wishes to 'improve' this solution, while minimal loss in the other objectives occurs. Clearly, the consideration whether the former or the latter solution is better matching the decision-makers preferences requires instantiation of the new solution in VR, too.

However, in complex problems the amount of solutions a decision maker needs to consider may be high in order to approach to his favourite solution, so that it becomes desirable to start the exploration from a solution among the Pareto solutions that is preferable in an unbiased sense. This is possible due to the involvement of fuzzy modelling in this work, as follows.

Although Pareto optimal solutions are equivalent in Pareto sense, it is noted that the solutions may still be distinguished. From figure 5, at the root node, the performance score is computed by the defuzzification process given by

$$w_1 f_1 + w_2 f_2 + w_3 f_3 = p \tag{8}$$

where f_1 is the output of the node *technical performance;* f_2 of node *utility performance*; f_3 of node *experiential performance.* That is, they denote the performance values for these aspects of the design, which are subject to maximization. The variable p denotes the design performance which is also requested to be maximized. In (32) w_1, w_2, and w_3 denote the weights associated to the connections from f_1, f_2 and f3 to the design performance. It is noted that $w_1+w_2+w_3=1$.

In many real-world optimization tasks the cognitive viewpoint plays an important role. This means it is initially uncertain what values $w_1, \ldots w_3$ should have. Namely, the node outputs $f_1, \ldots, f_3$ can be considered as the *design feature vector,* and the reflection of these features can be best performed if the weights w_1 ; ...; w_3 define the same direction as that of the feature vector. This implies that the performance p_{max} for each genetic solution is given by [19]

$$p_{\max} = \frac{f_1^{\,2} + f_2^{\,2} + f_3^{\,2}}{f_1 + f_2 + f_3} \tag{9}$$

Therefore, Eq. 9 is computed for all the design solutions on the Pareto front. Then the solution having maximal value of p_{max} is selected among the Pareto solutions. This way the particular design is identified as a solution candidate with the corresponding $w_1, w_2, \ldots, w_n$ weights. These weights form a priority vector $\boldsymbol{w}$*. If for any reason this candidate solution is not appealing, the next candidate is searched among the available design solutions with a desired design feature vector and the relational attributes, i.e., $w_1, w_2, \ldots, w_n$. One should note that, although performance does not play role in the genetic optimization, Pareto front offers a number of design options with fair performance leaving the final choice dependent on other environmental preferences. Using Eq. 9, second-order preferences are identified that are most promising for the task at hand, where ultimately maximal design performance is pursued.

To this end, to make the analysis explicit we consider a two-dimensional objective space . In this case, Eq. 9 becomes [15]

$$p = \frac{f_1^2 + f_2^2}{f_1 + f_2} \tag{10}$$

which can be put into the form

$$f_1^2 + f_2^2 - pf_1 + pf_2 = 0 \tag{11}$$

that defines a circle along which the performance is constant. To obtain the circle parameters in terms of performance, we write

$$f_1^2 + f_2^2 - pf_1 + pf_2 \equiv (x - x_1)^2 + (y - y_1)^2 - R^2 \tag{12}$$

From Eq. 12 we obtain the center coordinates x_1, y_1 and the radius R of the circle in terms of performance as

$$\begin{aligned} x_1 &= p/\ 2 \\ y_1 &= p/\ 2 \\ R &= p/\ \sqrt{2} \end{aligned} \tag{13}$$

The performance circle with the presence of two different Pareto fronts are schematically shown in figure 9a. From this figure, it is seen that the maximum performance is at the locations where either of the objectives is maximal at the Pareto front. If both objectives are equal, the maximal performance takes its lowest value and the degree of departing from the equality means a better performance in Pareto sense. This result is significant since it reveals that, a design can have a better performance if some measured extremity in one way or other is exercised. It is meant that, if a better performance is obtained, then most presumably extremity will be observed in this design. It is noted that the location of an expected superior Pareto optimal solution in this unbiased sense depends on the shape of the Pareto front, in particular on the degree of symmetry the Pareto front has w.r.t. the line passing from the origin of the objective space through the ideal point. This is illustrated in figure 9b, where it is seen that for a Pareto front that is asymmetrical w.r.t. to this diagonal a unique location of a solution with a superior performance may exist.

3.2 Implementation nr. 1

This implementation of the system in VR concerns the design of an interior space. The space is based on the main hall of the World Trade Centre in Rotterdam in the Netherlands. The aim is to optimally position a number of design objects in this space. The objects are a

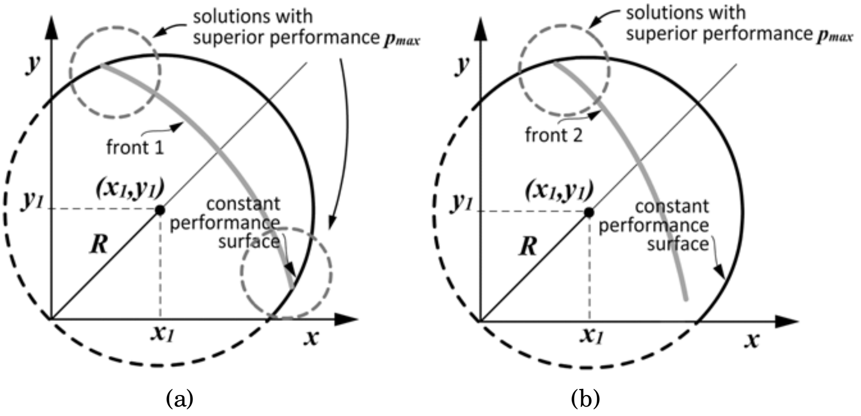


Fig. 9. Dependence of the location of desirable solutions on the shape of Pareto front

vertical building core hosting the elevators, a mezzanine, stairs, and two vertical ducts. The perception of a virtual observer plays a role in this task, because the objective involves a number of perception-based requirements. The function $f_x(x)$ shown in figure 10b is a probability density function (pdf) and given by Eq. 14 [20]. It models the visual attention of an *unbiased* virtual observer along a plane perpendicular to the observer's frontal direction. The unbiasedness refers that the observer has no a-priori preference for any particular direction within his visual scope over another one. Integral of the pdf over a certain length domain, i.e. of an object, yields perception expressed via a probability in this approach. The probability expresses the degree by which the observer is aware of the object.
The implementation of this model in virtual reality using a virtual observer termed *avatar* is illustrated in figure 11. From the figure it is noted that the avatar pays attention to the objects in the space equally in all directions in his visual scope. This is illustrated by means

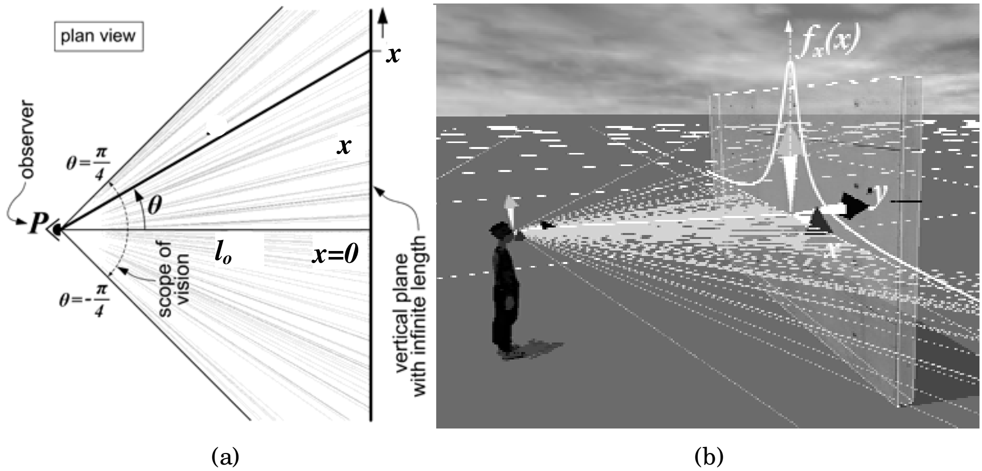


Fig. 10. Probabilistic perception model for a basic geometric situation, where the probability density $f_x(x)$ models visual attention along a plane object. Plan view (a); perspective view (b)

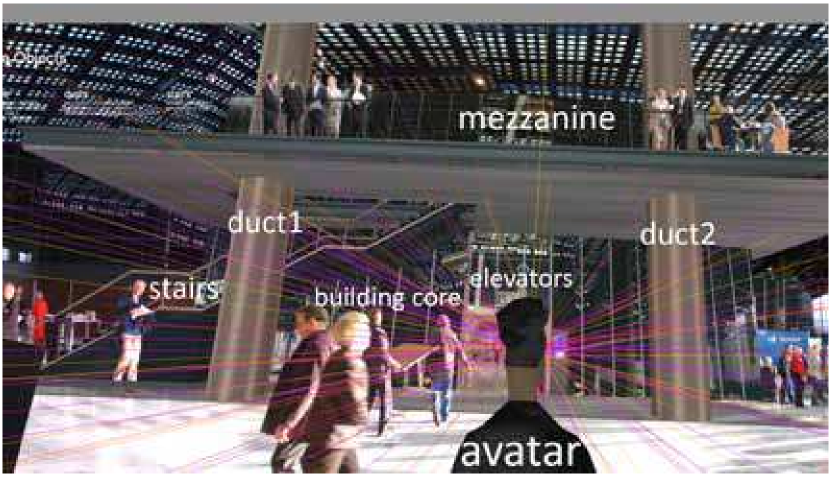


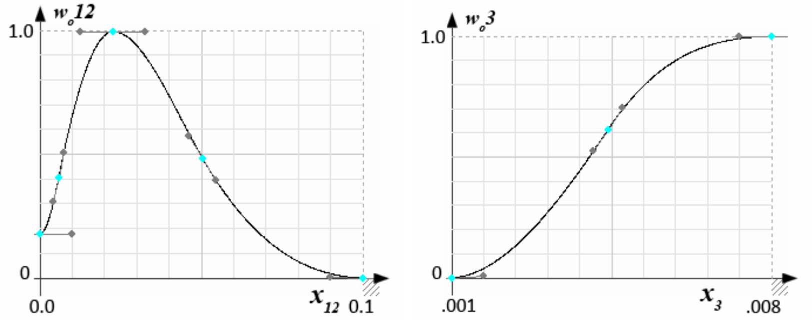
Fig. 11. Perception measurement by means of an avatar in virtual reality based on a probabilistic theory of perception

of the rays sent from the eyes of the avatar in random directions and intersecting the objects in the scene. The randomness has a uniform probability density w.r.t. the angle θ in figure 10a. In virtual reality implementation the amount of rays impinging on an object are counted and averaged in real time to approximate the perception expressed by a probability.

$$f_x(x) = \frac{2}{\pi}\frac{l_o}{\left(l_o^{\,2} + x^2\right)}\,, \quad -l_o < x < l_o \tag{14}$$

The perception model requires instantiation of objects to obtain the probability quantifying perception. That is, the GA determines the position of the objects, however their geometric extent is responsible for the perception of the observer. So, once a candidate scene is instantiated in virtual reality, the perception computations involving the geometric features of the scene objects are executed.

The results from the perception measurement are probabilities associated with the objects of the scene, that indicate to what extend an object comes to the awareness of an observer paying unbiased visual attention to the scene. This crisp information needs to be further evaluated with regards to the satisfaction of the goals at hand. The present design task involves several perceptual requirements. Two of them are shown in figure 12 as examples. One example is that the stairs should not be very noticeable from the avatar's viewing position, in order to increase the privacy in terms of access to the mezzanine floor. At the same time the stairs should not be overlooked too easily for people who do need to access the mezzanine floor. This is seen from the mf in figure 12a, where x_{12} denotes the perception degree and w_o12 denotes the fuzzy membership degree. A second example is that the elevators should be positioned in such a way that they are easily noticed from the avatar's viewing position, so that people who wish to access the office floors above the entrance hall easily find the elevators. This requirement is expressed by means of the fuzzy membership function in figure 12b, where increasing perception denoted by x_3 yields increasing membership degree w_o3.



(a) (b)

Fig. 12. Two requirements subject to satisfaction: perception of the stairs (a) and elevators (b)

It is noted that the perception computation using the probabilistic perception model yields x_{12} in figure 12b. The task is to optimally place the design objects satisfying a number of such perception requirements, and also some functionality requirements. The functionality requirements concern for instance the size of the space, which is influenced by the position of the building core object. The elemental requirements and their relation with the ultimate goal are seen from the fuzzy neural tree structure shown in figure 13. From the structure we note that the performance of the entrance hall depends on the performance of the design objects forming the scene. From this we note that the amount of objectives to be maximized is four, namely the outputs of nodes 4-7, whereas the elemental requirements total an amount of 12.

Figure 14 shows the results from the relaxed Pareto ranking approach. It is noted that the objective space has four dimensions, one for the performance of every design object. The representation is obtained by first categorizing the solutions as to which of the four

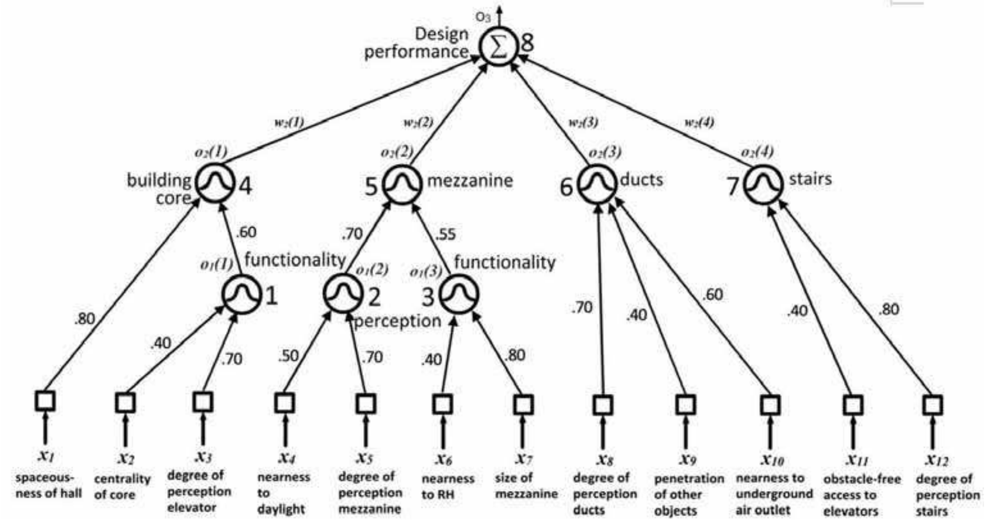

O_3
Design performance
8
$w_2(1)$
$w_2(2)$
$w_2(3)$
$w_2(4)$
$o_2(1)$
$o_2(2)$
$o_2(3)$
$o_2(4)$
building core
4
5
mezzanine
6
ducts
7
stairs
.60
.70
.55
$o_1(1)$
functionality
$o_1(2)$
$o_1(3)$
functionality
1
2
3
perception
.80
.40
.70
.50
.70
.40
.80
.70
.40
.60
.40
.80
x_1 spaceousness of hall
x_2 centrality of core
x_3 degree of perception elevator
x_4 nearness to daylight
x_5 degree of perception mezzanine
x_6 nearness to RH
x_7 size of mezzanine
x_8 degree of perception ducts
x_9 penetration of other objects
x_{10} nearness to underground air outlet
x_{11} obstacle-free access to elevators
x_{12} degree of perception stairs

Fig. 13. Neural tree structure for the performance evaluation

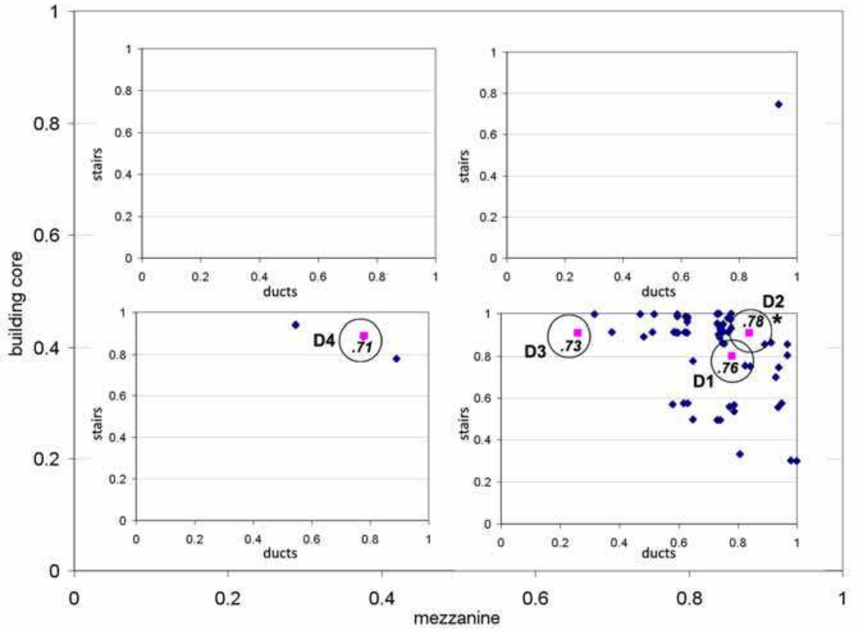


Fig. 14. Pareto optimal designs with respect to the four objective dimensions using relaxed Pareto ranking

quadrants in the two-dimensional objective space formed by the building core and mezzanine performance they belong, and then representing in each quadrant a coordinate system showing the stairs and ducts performance in this very quadrant. This way four dimensions are represented on the two-dimensional page.

Two Pareto optimal designs are shown in figures 15 and 16 for comparison. The maximal performance score as well as the performance feature vector for these solutions is shown in Table 1.

	core	mezzanine	ducts	stairs	p_{max}
D2	0.27	0.73	0.83	0.93	0.78
D4	0.48	0.49	0.78	0.89	0.71

Table 1. Performance of design D2 versus D4

From the table it is seen that design D2 outperforms design D4 with respect to the maximal performance p_{max} obtained using Eq. 9. It is also noted that the performance of D4 as to its features varies less compared to D2. The fact that D2 has a greater p_{max} confirms the theoretical expectation illustrated by figure 9 that solutions with more extreme features generally have a greater maximal performance compared to solutions with little extremity. The greatest absolute difference among D2 and D4 is the performance of the mezzanine. In D2 the mezzanine is located closer to associated functions, and this turns out to be more important compared to the fact that D4 yields more daylight on the mezzanine. Therefore D2 scores higher that D4 regarding the mezzanine. Additionally D2 slightly outperforms D4

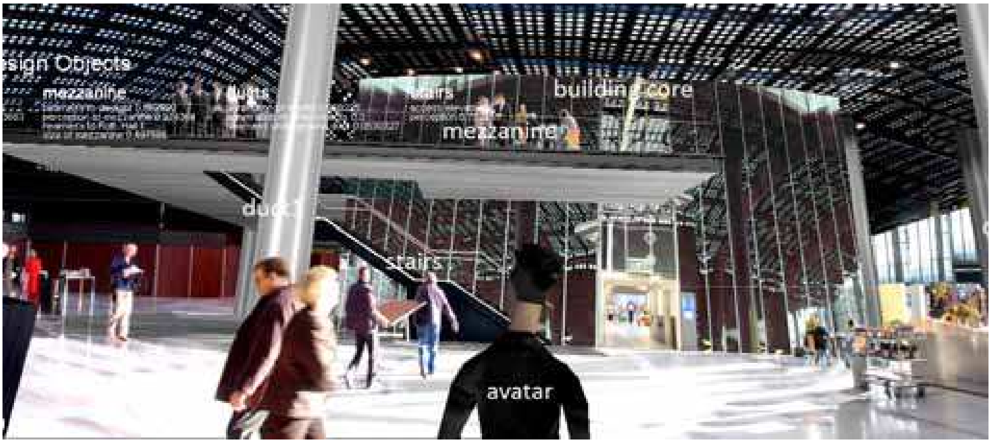


Fig. 15. Pareto-optimal design *D2* in Figure 14

regarding the performance of the ducts. This is because the ducts do not penetrate the mezzanine in D2, whereas in D4 they do. The latter is undesirable, as given by the requirements. Regarding the building core D2 is inferior to D4, which is because the spaciousness in D4 is greater and also the elevators are located more centrally. Regarding the stairs' performance, the difference among D2 and D4 is negligible. The latter exemplifies the fact that an objective may be reached in different ways, i.e. solutions that are quite different regarding their physical parameters may yield similar scores as to a certain goal. In the present case the greater distance to the stairs in D2 compared to D4 is compensated by the fact that the stairs is oriented sideways in D2, so that the final perception degree is almost the same. It is noted that D2 is the solution with the greatest maximal performance p_{max}, so that from an unbiased viewpoint it is the most suitable solution among the Pareto optimal ones. This solution is most appealing to be selected for construction. This result is an act of machine cognition, as it reveals that pursuing maximal performance in the present

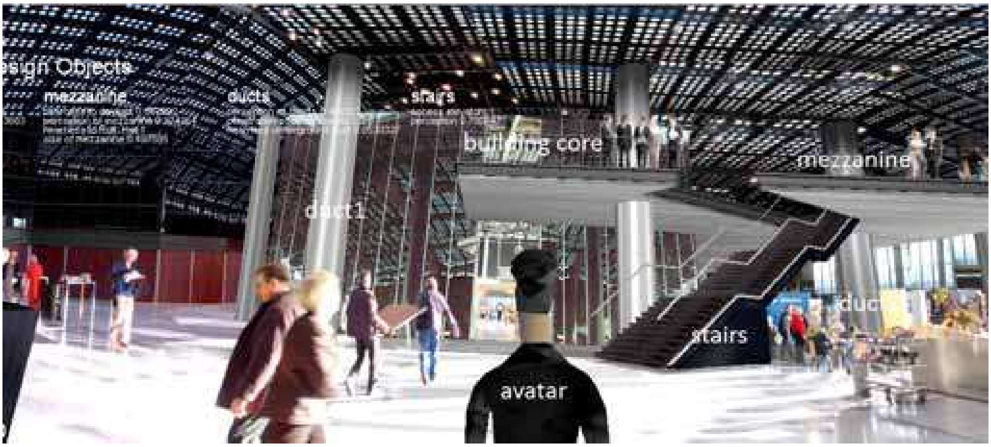


Fig. 16. Pareto-optimal design *D4* in Figure 14

task the stairs and ducts are more important compared to the building core from an unbiased viewpoint. This information was not known prior to the execution of the computational design process. It is interesting to note that the solution that was chosen by a human architect in a conventional design process without computational support was also similar to solution D2. The benefit of the computational approach is that it ensures identification of most suitable solutions, their unbiased comparison, and precise information on their respective trade-off as to the abstract objectives. This is difficult to obtain using conventional means. The diversity of solutions along the Pareto front, which is due to the relaxation of the Pareto concept is significant especially in order to facilitate the process of ensuing validation.

3.3 Implementation nr. 2

In the second implementation of the computational design system, object instantiation in VR is used for evaluation of solutions in a layout problem of a building complex for a performance measurement involving multiple objectives. In this task the spatial arrangement of a number of spatial units is to be accomplished in such a way that three main goals are satisfied simultaneously. These goals are maximizing the building's functionality and energy performance, as well as its performance regarding form related preferences. It is noted that the spheres shown in ensuing figures represent the performance of a number of alternative solutions for the three objectives of the design task.

The building subject to design consists of a number of spatial units, referred to as design objects, where every unit is designated to a particular purpose in the building. The task is to locate the objects optimally on the building site with respect to the three objectives forming suitable spatial arrangements. The objects are seen from figure 17 and their properties, which play role during the fitness evaluation of solutions generated by the algorithm, are given in table 3.

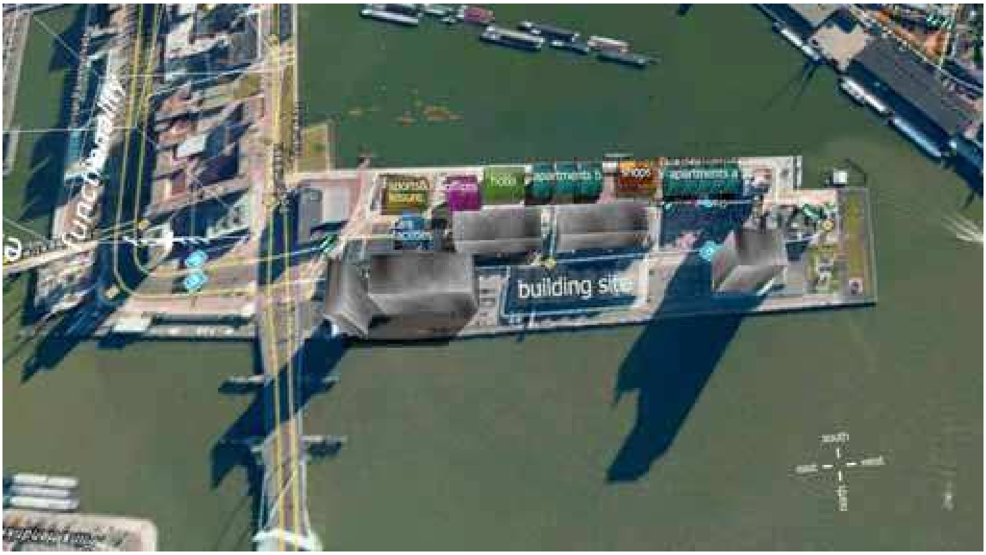


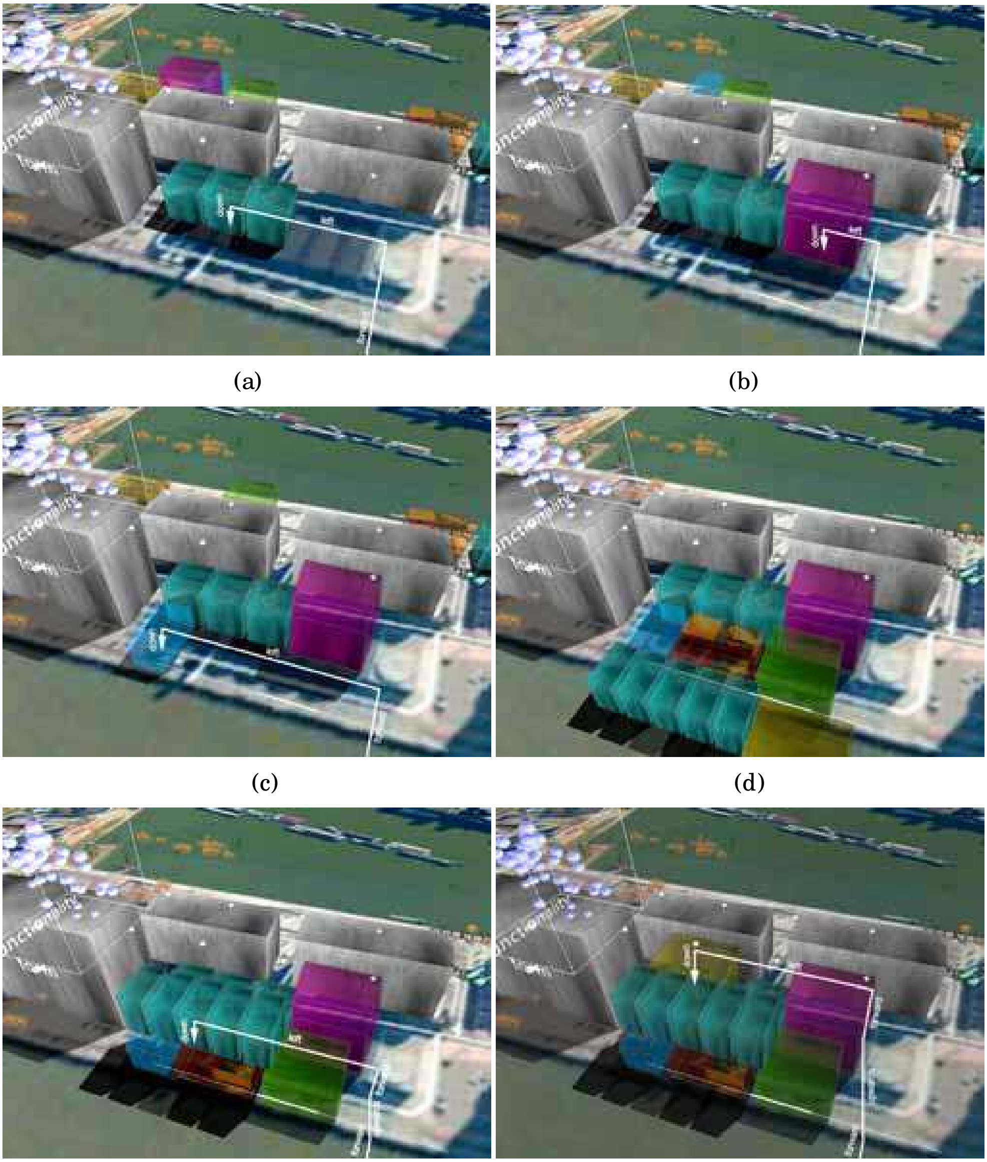
Fig. 17. Design objects subject to optimal positioning on the building site

	floorsurface (m²)	ceiling height (m)	Specific power q_I of inner heat sources (W/ m²)	surface amount of glass in façades (%)
apt_a_1	22000	2.7	2.1	30
apt_a_2	22000	2.7	2.1	30
apt_a_3	18500	3.2	2.1	40
apt_a_4	22000	2.7	2.1	30
apt_a_5	22000	2.7	2.1	30
apt_b_1	45000	2.7	2.1	30
apt_b_2	37000	3.2	2.1	40
apt_b_3	45000	2.7	2.1	30
hotel	74000	3	2.1	40
care	32000	3	2.1	20
shops	34000	5	4	50
offices	115000	3.5	3.5	70
sport	28000	6	3.5	70

Table 3. Properties of the design objects

The attributes given in Table 1 play an important role in particular in the evaluation of the energy performance of the solutions, which is described in the Appendix A. It is noted that the site is located in Rotterdam in the Netherlands, so that climate data from this location is used in the energy computations. It is further noted that the energy computations require information of the insulation value of the facades expressed by the U-value of the walls, U-value of windows and glass façade, as well as the g-value of the glass. In this task the U-value of the walls is 0.15 W/ m^{2}K; U-value of windows is 1.00W/ m^{2}K; and g-value of the glass is 0.5.

In order to let the computer generate a building from the components shown in figure 17, i.e. for a solution to be feasible, it is necessary to ensure that all solutions have some basic properties. These are that spaces should not overlap, and objects should be adjacent to the other objects around and above; also the site boundaries should be observed, in particular on the ground floor to permit pedestrian traffic along the waterfront. This is realized in the present application by inserting the objects in a particular sequential manner into the site. This is illustrated in figure 18. Starting from the same location, one by one the objects are moved forward, i.e. in southern direction, until they reach an obstacle. An obstacle may be the site boundary or another object previously inserted. When they touch an object they change their movement direction from the southern to the eastern direction, moving east until they again reach the site boundary or another object. As a final movement step the object will move down until it touches the ground plane, which is in order to account for different heights the objects have. Packing objects in two dimensions in this way is known as *bottom-left two heuristic packing routine* in literature, e.g. [21]. After the final object has been placed in this way, due to the fact that the sum of the objects' groundplanes exceeds the available surface on the site, some objects will overlap the site boundary or be situated entirely outside the site boundary. This is illustrated in figure 18d, where in the present example two building units – *apartments a* and *sports & leisure* are located outside of the southern site boundary. The boundary is indicated by means of a white line in the figure.



(a) (b)

(c) (d)

(e) (f)

Fig. 18. Generation process of a solution through sequential insertion of the design objects in 3D

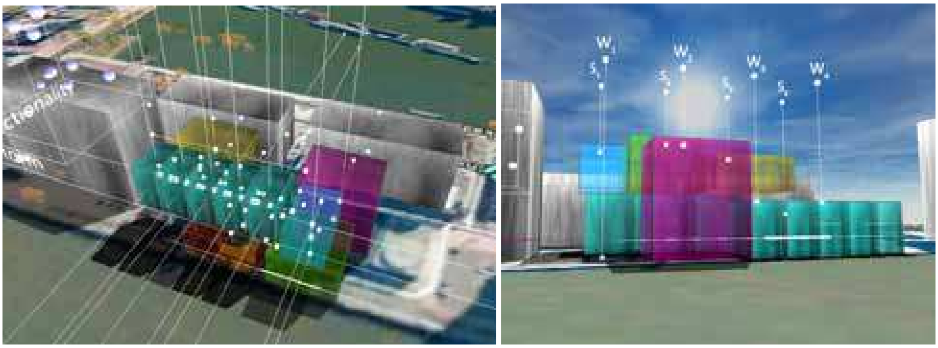
The objects exceeding the site will be inserted using a second movement procedure, where first the object is moved forward until it reaches an obstacle; then it is moved upwards until it reaches an upper boundary for the building, which is set to 140m and not visible in the figures. Then the object is moved forward again, until it touches an obstacle. Thereafter it is moved in eastern direction until touching an obstacle, and then down, so that it comes to rest on top the building below it.

It is noted that the decision from which side to insert the building components, and which location to use as the starting point for insertion is a matter of judgement, and it will strongly influence the solutions obtained. The insertion used in this application is due to the preference of the architect is to have the objects line up along the street, which is in northern boundary of the site.

In this task object instantiation is required for several reasons. One of the reasons can be already noted considering the above insertion process during the generation of feasible solutions. Namely, during the movement of an object into the site it is formidable to establish a formalism that can be used to predict the exact geometric condition of the configuration that is already found on the site when the object moves into it. The reason is that the amount of possible geometric configurations is excessive due to the amount of objects and also due to the fact that two of the objects, namely the offices and the hotel unit are permitted to have different amounts of floor levels, which is a parameter in the GA. As the floor surface amount is requested to remain constant, consequently both the object's height and floor plan is variable for these two objects.

Effectively, the spatial configuration an object will encounter during its insertion into the scene can only be known through execution of the insertion process, i.e. through instantiation of the objects on site as well as letting objects move into the site and testing for collisions during the movement. In this respect it is noted that the accuracy of placement is subject to determination, where the step length of the movement at every time frame during object insertion should be set to a small value, however not too small to avoid that the collision detection routine is called excessively. Next to the need for VR during this solution generation procedure, the instantiation is needed to execute the measurements indicated by the letter *m* in figure 2 as follows.

For the evaluation of the energy performance of the building it is necessary to compute the *transmission heat loss* denoted by Q_T [22]. Q_T quantifies how much energy will be lost through the facades of every building component over the period of one year due to temperature difference between inside and outside air temperature. In order to obtain this value it is necessary to verify for every façade of a building unit, whether it is adjacent to another building component, or adjacent to outside air. Also it is necessary to compute the solar gain Q_S, which quantifies the amount of solar energy that penetrates into the building unit through the glazing of the facades. For a certain façade surface, $\mathrm{Q_S}$ depends, among other factors, on the distance from another building unit located in front of the facade causing a shadow. Therefore, to accomplish computation of $\mathrm{Q_T}$ and $\mathrm{Q_S}$ it is necessary to measure if another object is adjacent to the façade in question, located in front of the facade at some distance close enough to cause a shadow, or if there is no object in front of the façade causing a shadow on it. For this purpose a test procedure is executed in the virtual reality, where for every façade the distance to objects in front of it is measured. It is clear that this test requires object instantiation due to the manifold possible geometric configurations in the search. The test is executed by means of rays that are emitted from the centre point of the building component in question and the intersection with other objects is detected. This is shown in figure 19a. The resulting information is then used in the computations of Q_T and Q_S in order to compute the heat energy Q_H required to heat the building over the period of one year per m^2 of floor surface area. The output Q_H is the result from energy computations using a steady state model given in the Appendix. From the neural tree in figure 20 it is seen that the energy performance evaluation involves a single fuzzy membership function, i.e. it does not involve inner nodes.



(a) (b)

Fig. 19. Verification of thermal environment by means of ray intersection tests for heat loss computation (a); measurement of heights of the building for estimating the satisfaction of form preferences

The membership function is shown in figure 21a, where it is seen that the input information for the energy evaluation is the heat energy Q_H expressed as energy per m^2 of floor surface area and per year. From figure 21a we note that the satisfaction of the energy requirement increases with decreasing energy, and that the satisfaction, expressed by the membership degree μ reaches its maximum for heat energy consumption below 2.2 kWh/ m^{2}a, and satisfaction diminishes for energy amount beyond 4.4 kWh/ m^{2}a. It is noted that this range concerns relatively low amount of energy compared to most contemporary building projects. This mainly due to the large size of the building units, where the amount of exterior surfaces with respect to the floor is relatively small.

For the evaluation of the performance regarding form preferences for the building, object instantiation in VR is required in order to execute other measurements. This is shown in figure 19b. From the figure it is seen that from 8 locations above the building test rays are sent downwards to measure the building's height at these locations. This information is used to compute to what extend the shape of the building satisfies some form preferences of the architect. The form preferences are seen from the fuzzy neural tree shown in figure 20.

The evaluation of the form preferences has two major aspects, the first one concerns the variations of heights in the building's skyline; the second one concerns the average height of the building. For both aspects two sub-aspects are distinguished in the model: the situation along the side facing the street (along the southern site boundary), and the side along the waterfront (along the northern boundary). For the height variation assessment, the difference in height measured between two adjacent measurement points S_n or W_n is obtained using the ray-tracing in VR seen from figure 19b. This difference is used as input in the membership function shown in figure 21b. From the membership function it is seen that the height variation is demanded to be rather large, i.e. the architect aims for a non-monolithic shape of the building, so that it is deemed to express what may be termed as a *playful* looking shape. This is seen from the maximum of the membership function being located at about 76m of height difference.

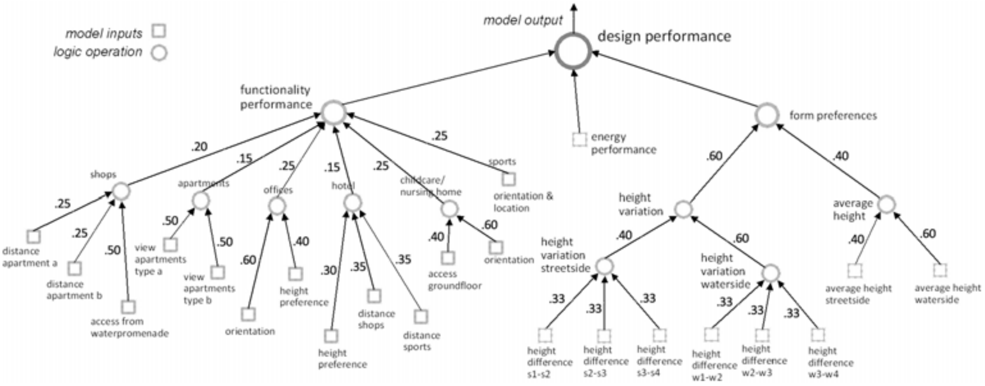
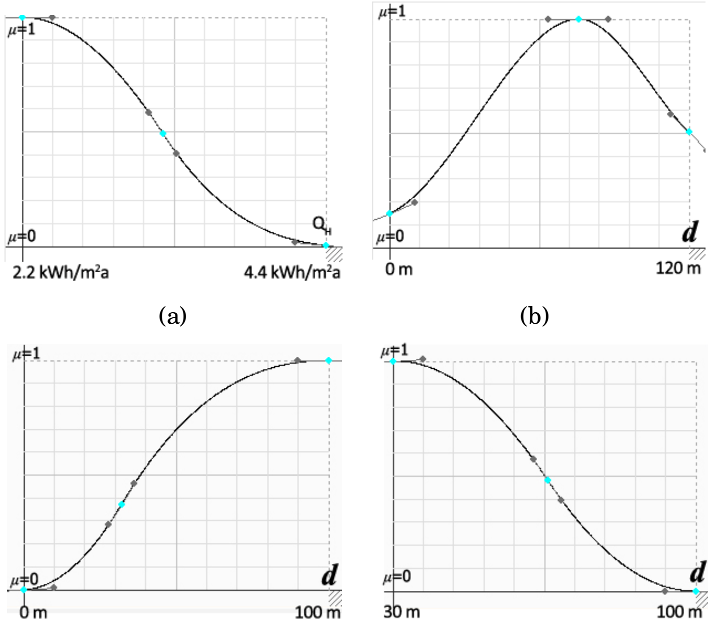


Fig. 20. Fuzzy neural tree for performance evaluation of the candidate solutions



(a) (b)

(c) (d)

Fig. 21. Fuzzy membership functions used for energy performance evaluation (a); for evaluation of the height variation in the building's skyline (b); for evaluation of the average height along the street-side (c); along the waterfront (d)

Concerning the requirements on average height of the building the architect prefers to have a high average height along the street side, and a low average height at the waterfront. This is to emphasize the urban character of the street, whereas the lower height along the waterfront is to give the building a more accessible expression when perceived by people walking along the waterfront. The requirement for a high average height along the street-side is seen from the membership function in figure 21c, yielding maximum membership degree at 100m and diminishing as the height reduces. The requirement for a low average height along the water-front is seen from the membership function in figure 21d, where the membership degree diminishes with increasing height.

In the same way, during evaluation of a design alternative the tree is provided with input values obtained from the virtual building instantiated in VR, and the fuzzification processes are carried out at the terminal nodes. The fuzzification yields the degree of satisfaction for the elemental at the terminal nodes of the neural tree.

The root node of the neural tree shown in figure 20 describes the ultimate goal subject to maximization, namely the design performance and the tree branches form the objectives constituting this goal. The connections among the nodes have a weight associated with them, as seen from the figure. In the same way as the membership functions at the terminals, the weights are given by a designer as an expression of knowledge, and the latter specify the relative significance a node has for the node one level closer to the root node. In particular the weights connecting the nodes on the penultimate level of the model indicate how strongly the output of these nodes influences the output at the root node. It is noted that in the multi-objective optimization case the latter weights are not specified a-priori, but they are subject to determination after the optimization process is accomplished.

The fuzzified information is then processed by the inner nodes of the tree. These nodes perform the AND operations using Gaussian membership functions as described above, where the width-vector of the multi-dimensional Gaussian reflects the relative importance among the inputs to a node. Finally this sequence of logic operations starting from the model input yield the performance at the penultimate node outputs of the model. This means the more satisfied the elemental requirements at the terminal level are, the higher the outputs will be at the nodes above, finally increasing the design performance at the root node of the tree. Next to the evaluation of the design performance score, due to the fuzzy logic operations at the inner nodes of the tree, the performance of any sub-aspect is obtained as well. This is a desirable feature in design, which is referred to as *transparency*

The multi objective optimization is accomplished using a multi-objective genetic algorithm with adaptive Pareto ranking. It is used to determine the optimal sequence of insertion, so that the three objectives are maximally fulfilled. Every chromosome contains the information for every object, at which rank in the insertion sequence it is to be inserted, as well as the information for the relaxation angle to use during the Pareto ranking for the particular solution. It is noted that the information a chromosome contains in order to determine the sequence of insertion is in the form of float numbers, where one float number is assigned to every object. The objects are then sorted based on the size of the float numbers, so that an object with a higher number will be inserted before one with a lower number. Using float numbers in the chromosome, as opposed to e.g. an integer number denoting a unique sequence of insertion, allows a genetic algorithm with conventional crossover procedure to generate more suitable solutions from the genetic combination of two successful ones. This is because the float number sequence is unbiased with respect the objects to be inserted, whereas an integer coding of the sequences has an inherent bias making it necessary to reflect this bias in the crossover procedure.

The performance evaluation model is used during the evolutionary search process aiming to identify designs with maximal design performance. In the present case we are interested in a variety of alternative solutions that are equivalent in Pareto sense. The design is therefore treated as a multi-objective optimization as opposed to a single-objective optimization. In single-objective case exclusively the design performance, i.e. the output at the root node of the neural tree, would be subject to maximization. In the latter case, the solution would be the outcome of a mere convergence and any cognition aspect would not be exercised. In the multi-objective implementation the outputs of the nodes *functionality*, *energy*, and *form preferences*, which are the penultimate nodes, are subject to maximization. Their values are used in the fitness determination procedure of the genetic algorithm. Employing the fuzzy neural tree in this way the genetic search is equipped with some human-like reasoning capabilities during the search. The part of the tree beyond the penultimate nodes is for the de-fuzzification process, which models cognition, so that ultimately the design performance is obtained at the root node.

3.4 Application results

To exemplify the solutions on the Pareto front, three resulting Pareto-optimal designs D1-D3 are shown in figures 22-24 respectively. In the left part of the figures the location of the particular solution in the three-dimensional objective space is seen together with the locations of the other solutions.

The solutions in objective space are represented by spheres. The size of the sphere indicates the maximal performance value of the corresponding solution. That is, a large sphere indicates a high maximal design performance, and conversely a small sphere indicates a low performance.

Design *D1* is the design among the Pareto solutions having the highest maximal design performance, as obtained by Eq. 9, namely $p=.75$. It has a high *energy* and *form* performance, namely .76, and .88 respectively, while its *functionality* performance is moderate, being only .50. The high performance as to *form* is due to the strong variations of building-height along the building's skyline and the lower water-front versus higher street side, which match to the requirements. The low functionality performance is mainly due to low performance of office and childcare facilities, where the offices are expected to be a tall building-unit and offer a good view of the waterside.

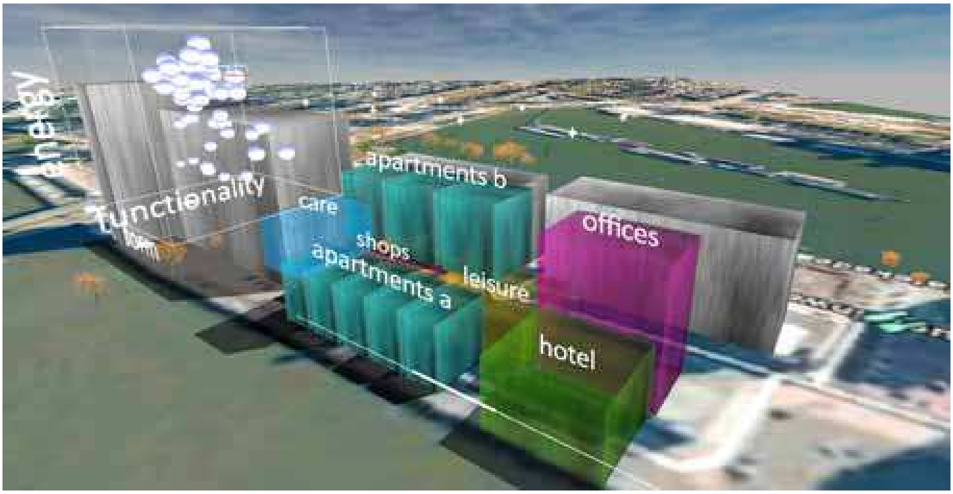


Fig. 22. Design D1 having a p_{max} of .75 being the highest among the Pareto solutions

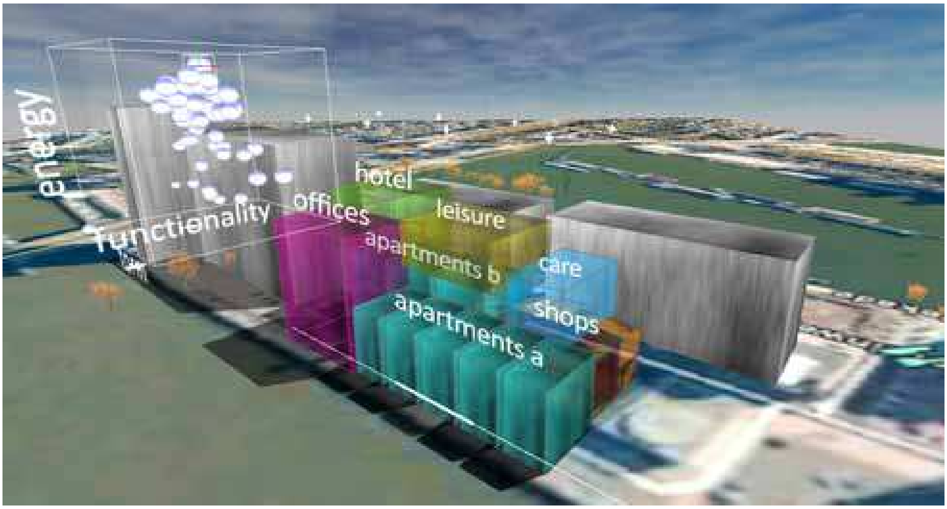


Fig. 23. Design D2 having a high energy performance

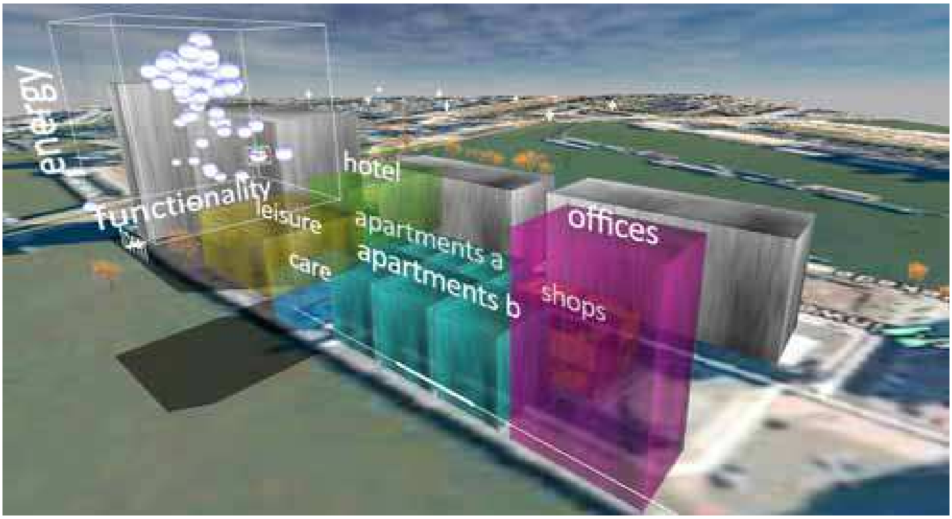


Fig. 24. Design D3 having a high functionality performance

Design *D2* has the highest *energy* performance among the Pareto solutions (.91) while *form* and *functionality* are moderate (.55 and .47). Its maximal design performance is *p=.70*. The high energy performance is due to the very compact overall shape, and also due to the fact that the office building, having a large amount of glazing percentage, has a compact shape implying few energy loss.

Design *D3* has a high functionality performance (.81), while energy performance is low (.23) and form performance is moderate (.41). Its maximal design performance is *p=.61*. The functionality performance is high, because the requirements for office, shops and care are highly satisfied. The energy performance is low, because the overall building is not compact and most of the envelope of the office building component is exposed to outside air, which yields undesirably high heat energy consumption of the building.

From the results we note that design *D1* has a maximal performance that is higher than for the other Pareto optimal designs described by factor 1.07 and 1.23 respectively. That is, D1 clearly outperforms the other designs regarding their respective maximal performance. This means that when there is no a-priori bias for any of the three objectives, it is more proficient to be less concerned with functionality, but to aim for maximal energy performance and form qualities in the particular design task at hand. That is, in absence of second-order preferences, design D1 should be built, rather than the other designs.

4. Conclusions

The role of object instantiation in virtual reality during a computational design process is described by means of a computational intelligence approach implemented in virtual reality. The approach establishes Pareto front in a multi-objective optimization involving a stochastic search algorithm and a fuzzy model of the design requirements. The instantiation of solutions in VR plays a necessary role in the search process, as it permits evaluating solutions with respect to abstract object features that are not readily obtained from the parameters subject to identification through the search. Next to its role during the search for optimality, VR also facilitates the selection process among the Pareto optimal solutions, and the process of validating the criteria used in the search, which is also exemplified. The necessary role of VR during the search for optimality is demonstrated in two applications from the domain of architectural design, where the object instantiation is needed for the effectiveness of several procedures during the search process. In one application it is required for execution of a measurement procedure to quantify perceptual qualities of the design objects involving a virtual observer. In this task optimal positioning of a number of interior elements is obtained satisfying perceptual and functionality related requirements. In the second application instantiation in VR is required to facilitate the solution generation using a two heuristic packing strategy. Next to that it is needed in this application in order to permit measurement of functionality, energy, and form related performance of the solutions. A building consisting of several volumes is obtained, where these objectives are maximally satisfied. This is accomplished by identifying an optimal sequence of arranging the volumes, so that the three objectives pertaining are satisfied. In both applications the linguistic nature of the requirements is treated by using a fuzzy neural tree approach that is able to handle the imprecision and complexity inherent to the concepts, forming a model. This model plays the role of fitness function in the adaptive multi-objective evolutionary search algorithm, so that the search process is endowed with some human-like reasoning capabilities. The involvement of a fuzzy model requires the crisp input information for fuzzification and further processing via the fuzzy model. This is provided through the instantiation of objects and ensuing measurements in virtual reality. With this understanding VR can be considered to act as interface between the domain of quasi physical object features and the domain of abstract goals during the search for optimality.

Appendix - Energy Computations

The input of the fuzzy membership function expressing the energy performance shown in the neural tree in figure 21a requires as input the energy demand for heating over the period of one year and per floor surface area. This value is denoted by Q_H and given in the unit kWh/ m^{2}a. The size of the floor surface areas are given in Table 1. Q_H is computed as follows [22].

$$Q_H = Q_L - Q_G \tag{A1}$$

where Q_L denotes the sum of the energy losses and Q_G denotes the sum of energy gains of the building unit. Let us first consider the losses:

$$Q_L = Q_T + Q_V \tag{A2}$$

In Eq. A2 Q_T denotes losses through transmission via the building envelope, and Q_V denotes losses through ventilation. Q_T is computed by for every façade element n delimiting the unit as given by

$$Q_T = \sum_n A_n \cdot U_n \cdot f_t \cdot G_t \tag{A3}$$

where A_n denotes the surface amount of the n-th façade element in m^2; U$_n$ denotes the U-value of the façade element given in the unit W/ m^{2}K; f_t denotes a temperature factor to account for reduced losses when a façade is touching the earth (.65) versus the normal condition of outside air (1.0); G_t denotes the time-integral of the temperature difference between inside and outside air temperature given in the unit kKh/ a. In this implementation G$_t$=79.8 kKh/ a.
Q$_V$ is computed for a building unit by

$$Q_V = V \cdot n_V \cdot c_{air} \cdot G_t \tag{A4}$$

where V denotes the air volume enclosed within the unit given in m^3; n_V denotes the energetically effective air exchange rate of the ventilation system during the heating period given in the unit 1/ h, which is n_V=0.09/ h in this implementation; c_{air} denotes the heat capacity of air c_{air}=0.33 Wh/ m^{3}K.
Considering the energy gain: Q_G is obtained by

$$Q_G = \eta_G \cdot Q_F \tag{A5}$$

where η_G is a factor denoting the effectiveness of the heat gains, and Q_F denotes the free heat energy due to solar radiation and internal gains, given by

$$Q_F = Q_S + Q_I \tag{A6}$$

where Q_S denotes the gain due to solar radiation and Q_I denotes the internal gain:

$$Q_S = \sum_n f_r \cdot g_w \cdot A_{n,w} \cdot G_d \tag{A7}$$

In Eq. A7 for the n-th façade of a building unit f_r denotes a reduction factor that models the effect of a shadow on the façade. In the present implementation this factor is computed online using a measurement in VR. The factor g_w in Eq. A7 denotes the *g-value* of the window glazing used in the façade. This value expresses the total heat energy flux rate permitted through the glass. In the present case g_w=0.5. $A_{n,w}$ denotes the amount of window surface in the façade in m²; G_d denotes the direction dependent solar radiation energy given in the unit kWh/ m²a. In the present climatic situation G_{south}=321 kWh/ m²a; G_{north}=145 kWh/ m²a; G_{east}=270 kWh/ m²a and G_{west}=187 kWh/ m²a.

Q_I in Eq. A6 is given by

$$Q_I = 0.024 \cdot t \cdot q_I \cdot A_f \tag{A8}$$

where the number 0.024 is a conversion factor having the unit kh/ d; t denotes the length of the heating period in days. In the present case t=205d. P_S denotes the specific power q_I of inner heat sources like people, lighting, computers, etc. given in the unit W/ m². For the different building units subject to positioning in this task the different values for q_I are given in Table 3. The factor η_G in Eq. A5 is obtained by

$$\eta_G = \frac{1-\left(Q_F / Q_V\right)^5}{1-\left(Q_F / Q_V\right)^6} \tag{A9}$$

5. References

[1] L. J. Hettinger and M. W. Haas, Virtual and Adaptive Environments: Applications, Implications, and Human Performance Issues: Lawrence Erlbaum, 2003.

[2] S. K. Ong and A. Y. C. Nee, Virtual Reality and Augmented Reality Applications in Manufacturing. London: Springer, 2004.

[3] Y. Yang, L. Engin, E. S. Wurtele, C. Cruz-Neira, and J. A. Dickerson, "Integration of metabolic networks and gene expression in virtual reality," Bioinformatics, vol. 21, pp. 3645-3650, 2005.

[4] J. Blascovich, J. M. Loomis, A. C. Beall, K. R. Swinth, C. L. Hoyt, and J. N. Bailenson, "Immersive virtual environment technology as a methodological tool for social psychology," Psychological Inquiry, vol. 13, pp. 103-124, 2002.

[5] J. Whyte, Virtual Reality and the Built Environment. Oxford: Architectural press, 2002.

[6] M. S. Bittermann, "Intelligent Design Objects (IDO) - A cognitive approach for performance-based design," in Department of Building Technology. vol. PhD Delft, The Netherlands: Delft University of Technology, 2009, p. 235.

[7] J. A. Wise, V. D. Hopkin, and P. Stager, (eds.), "Verification and validation of complex systems: Human factors issues," in NATO Advanced Study Institute - ASI, Vimeiro, Portugal, 1992, p. 705.

[8] K. Deb, Multiobjective Optimization using Evolutionary Algorithms: John Wiley & Sons, 2001.

[9] C. A. C. Coello, D. A. Veldhuizen, and G. B. Lamont, Evolutionary Algorithms for Solving Multiobjective Problems. Boston: Kluwer Academic Publishers, 2003.

[10] E. J. Hughes, "Evolutionary many-objective optimisation: many once or one many?," in IEEE Congress on Evolutionary Computation CEC'2005, Edinburgh, Scotland, 2005, pp. 222-227.

[11] J. Horn, N. Nafploitis, and D. E. Goldberg, "A niched Pareto genetic algorithm for multiobjective optimization," in First IEEE Conf. on Evolutionary Computation, 1994, pp. 82-87.

[12] K. Deb, P. Zope, and A. Jain, "Distributed computing of pareto-optimal solutions with evolutionary algorithms," in 9th annual conference on Genetic and evolutionary computation London, England, 2003, pp. 532-549.

[13] J. Branke, T. Kaussler, and H. Schmeck, "Guiding multi-objective evolutionary algorithms towards interesting regions," AIFB University of Karlsruhe, Germany 2000.

[14] K. Deb, J. Sundar, U. Bhaskara, and S. Chaudhuri, "Reference point based multi-objective optimization using evolutionary algorithm," Int. J. Comp. Intelligence Research, vol. 2, pp. 273-286, 2006.

[15] Ö. Ciftcioglu and M. S. Bittermann, "Adaptive formation of Pareto front in evolutionary multi-objective optimization," in Evolutionary Computation, W. P. d. Santos, Ed. Vienna: In-Tech, 2009, pp. 417-444.

[16] A. Jaszkiewicz, "On the computational efficiency of multiple objective metaheuristics: The knapsack problem case study," European Journal of Operational Research, vol. 158, pp. 418-433, 2004.

[17] L. A. Zadeh, "Fuzzy logic, neural networks and soft computing," Communications of the ACM, vol. 37, pp. 77-84, 1994.

[18] O. Ciftcioglu, M. S. Bittermann, and I. S. Sariyildiz, "Building performance analysis supported by GA," in 2007 IEEE Congress on Evolutionary Computation, Singapore, 2007, pp. 489-495.

[19] M. S. Bittermann and O. Ciftcioglu, "A cognitive system based on fuzzy information processing and multi-objective evolutionary algorithm," in IEEE Conference on Evolutionary Computation - CEC 2009 Trondheim, Norway: IEEE, 2009.

[20] Ö. Ciftcioglu, M. S. Bittermann, and I. S. Sariyildiz, "Towards computer-based perception by modeling visual perception: a probabilistic theory," in 2006 IEEE Int. Conf. on Systems, Man, and Cybernetics, Taipei, Taiwan, 2006, pp. 5152-5159.

[11] E. Hopper and B. C. H. Turton, "An empirical investigation of metaheuristic and heuristic algorithms for a 2D packing problem," Eur. J. Oper. Res., vol. 128, pp. 34-57, 2001.

[22] W. Feist, E. Baffia, J. Schnieders, R. Pfluger, and O. Kah, Passivhaus Projektierungs Paket (PHPP) 2007. Darmstadt: Passiv Haus Institut, 2007.

Virtual Reality
Edited by Prof. Jae-Jin Kim

ISBN 978-953-307-518-1
Hard cover, 684 pages
Publisher InTech
Published online 08, December, 2010
Published in print edition December, 2010

Technological advancement in graphics and other human motion tracking hardware has promoted pushing "virtual reality" closer to "reality" and thus usage of virtual reality has been extended to various fields. The most typical fields for the application of virtual reality are medicine and engineering. The reviews in this book describe the latest virtual reality-related knowledge in these two fields such as: advanced human-computer interaction and virtual reality technologies, evaluation tools for cognition and behavior, medical and surgical treatment, neuroscience and neuro-rehabilitation, assistant tools for overcoming mental illnesses, educational and industrial uses In addition, the considerations for virtual worlds in human society are discussed. This book will serve as a state-of-the-art resource for researchers who are interested in developing a beneficial technology for human society.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821