Use of virtual reality and constraint programming techniques in interactive 3D objects layout

KEFI Marouene
Laboratoire d'Ingénierie des Systèmes Automatisés
Université d'Angers, 62 Avenue ND du Lac, Angers, France
marouene.kefi@univ-angers.fr

RICHARD Paul
Laboratoire d'Ingénierie des Systèmes Automatisés
Université d'Angers, 62 Avenue ND du Lac, Angers, France
paul.richard@univ-angers.fr

BARICHARD Vincent
Laboratoire d'Etudes et de Recherche en Informatique
Université d'Angers, 2 Bd Lavoisier, Angers, France
vincent.barichard@univ-angers.fr

ABSTRACT

In this paper, we propose innovative system in order to assist the user in a 3D objects layout context. Through a combination between virtual reality (VR) and constraint programming (CP) technique, user's 3D interaction and manipulation will be translated to incoming queries of a constraints solver which propagate constraints and generate a new possible solution. The computed solution is transmitted, as new positions of 3D objects, to virtual environment (VE) which reconfigures itself. We focus in this paper on the architecture of our system and we describe the implementation of several constraints and some first results.

Keywords

Virtual environment, 3D interaction, Decision-making, 3D objects layout, constraint programming.

1. INTRODUCTION

A spatial configuration problem can be defined as a placement problem for which, and while satisfying the constraints, a positioning of components inside the container is looked for. It has applications in many industrial sectors. Often solved by hand from intuition and experience of designers, the development of automatic methods to solve the problem becomes a challenge at time when systems become more complex.

VR is defined as a system composed of software and hardware elements stimulating a realistic human interaction with virtual objects which are synthetic models of real or imaginary objects. The 3D interaction is the major component of VR, it allows the user to be able to change the course of events in a synthetic environment [Bowman, 1999].

VEs technology is now recognized as a powerful design tool in industrial sectors such as manufacturing, process engineering, construction, and aerospace industries [Zorriassatine et al., 2003].

However, in many cases, VEs are being used as a pure visualization tool for assessing the final design. VR can be used in many contexts of decision making involving several constraints, such as 3D objects layout which can be a tedious and costly task.

Thus the classic use of VEs does not provide assistance to the user in a 3D layout context and does not furnish indication on the best positioning of 3D objects. The integration of an intelligent module (constraints solver in our case) in VEs could resolve the interactive spatial configuration problem.

The notion of constraint is naturally present in several areas such as resources allocation, planning and industrial production. We can define a constraint as a property or condition that must be satisfied, it can be expressed as a relationship or a restriction on one or more variables.

To provide a solution of 3D objects layout problem, we present an intelligent virtual environment allowing the user to interact with virtual objects while respecting the predefined constraints of design. From a set of 3D objects, the user can select those which will constitute the 3D scene and specify their geometric properties (dimensions, colors ...) and semantic ones (temperature, light, vibration ...).

2. RELATED WORK

Some works on the under constraints programs in VEs have been developed. For example, Xu et al., have treated the combination of physics, semantics, and placement constraints and how it permits to quickly and easily layout a scene [Xu et al., 2002]. The author generalized distributions and a richer set of semantic information leading to a new modeling technique where users can create scenes by specifying the number and distribution of each class of object to be included in the scene. Sanchez et al. have presented a general-purposed constraint-based system for non-isothetic 3D-object layout built on a genetic algorithm [Sanchez et al., 2002]. This system is able to process a complex set of constraints, including geometric and pseudo-physics ones.

More recently, Calderon et al., have presented a novel framework for the use of VEs in interactive problem solving [Calderon et al., 2003]. This framework extends visualization to serve as a natural interface for the exploration of configuration space and enables the implementation of reactive VEs.

It must be noted that these previous works are based on CLP and Prolog [Diaz and Codognet, 2001] or genetic algorithms. However, in the last few years, powerful CP-based solvers such as Gecode [Schulte, 1997] have been developed.

In spite of interest of previous works, they present some limits and can be extended in different directions. For instance, we envisage offering more interactivity (by using haptic feedbacks and stereoscopic images) to the user for more efficient object manipulation. In addition, and for more clarity, an explanatory information module will also be provided to justify the infeasibility of certain configurations proposed by the user.

3. SYSTEM DESCRIPTION

Our system is a 3D real-time environment based on CP techniques. It supports the resolution of interactive 3D objects layout. Through permanent communication, the choice of objects and constraints as well as user's 3D manipulation will be converted to queries sent to the solver. The work of the solver will be translated into automatic reconfiguration of VE (Figure 1). In addition, this system can present to the user many solutions (feasible spatial configurations) that will be able to explore by a specific device.

In order to intensify the user's immersion in the VE, a human-scale virtual reality platform is used in our first tests (shown in Figure 2 and described in the next section).

3.1 Architecture of interaction model

The aim of the interaction model is to make the correspondence between user's interactions with VE and inputs / outputs of the solver.

In our case, the work of the solver is based on a specific logic, depending on which, it is triggered by the addition of new constraints and it produces results in the form of new positions of objects. Thus, two aspects are concerned: (1) how the solver can respond to user's actions? (2) how the solver's results will interactively modify the VE?.

From a configuration of objects showed in the VE, the user can interact with it by moving some objects. This manipulation generates an event that will be used by the communications module (based on *threads*) to create new queries to the solver.

Acting according to these queries, the solver will produce new results sent directly to the virtual environment in order to update the current spatial configuration. Consider the simple example from an initial solution computed by the solver, the user moves the gray object (circled object) to the right (Figure 1). An event will be automatically generated from which the communication module "post" new constraints in the solver. These constraints will be applied on object which index is encapsulated in the event sent to the solver. Thus the solver will be re-called and the new position of the concerned object, and possibly those of other objects, will be encapsulated in another event sent to the VE (via the communication module) that extract new positions and reconfigure itself.

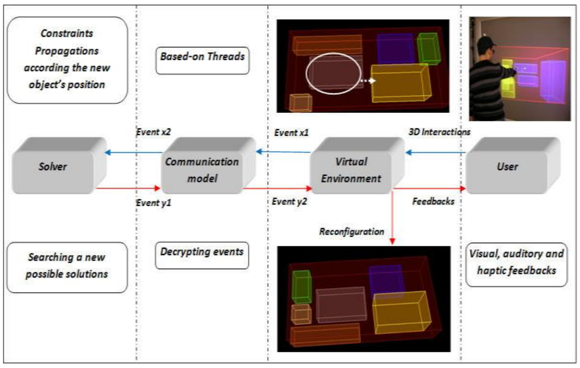

Figure 1. Architecture of intelligent VE.

4. VIRTUAL REALITY PLATFORM

The In order to intensify the user's immersion in the virtual world and assist him in his 3D arrangement, a human-scale virtual reality platform is used in our first tests (Figure 2). VIREPSE is a human-scale VE that provides force feedback using the SPIDAR system (Space Interface Device for Artificial Reality) [Richard et al., 2006]. Stereoscopic images are displayed on a rear-projected large screen (2m x 2.5m) and are viewed using polarized glasses. Four motors are placed on the corners of a cubic frame surrounding the user. By controlling the tension of each string, the system generates appropriate forces (Figure 2).

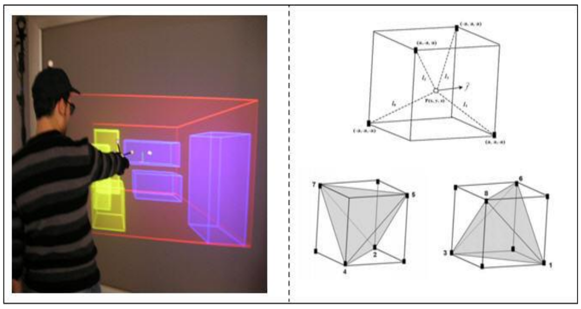

Figure 2. Multi-modal platform: VIREPSE

5. COMBINATION RV-CP

As mentioned before, our objective is to propose and implement interactive approach to solving interactive 3D layout problems [Kefi et al., 2010]. This approach is based on the architecture and interaction framework described above. From a GUI (Graphical User Interface), the user begins by selecting 3D objects to place and constraints to satisfy. Then the system will launch a dialogue with the solver to check the feasibility of the 3D arrangement. As illustrated in the next figure, the user cans interact with the proposed solution (computed by the solver) by moving its constituent objects. After each displacement, the solver is re-called to consider new constraints and calculate new solutions or cancel the last displacement (if at least one constraint is violated). Once the new solution computed, the 3D environment is informed of the new positions of objects and will automatically reconfigure itself (Figure 3).

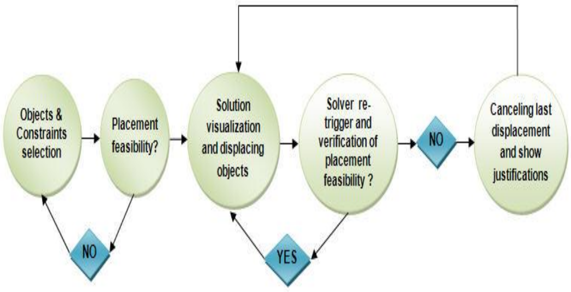

Figure 3. Illustration of the interactive approach

6. CONSTRAINTS IMPLEMENTATION AND FIRST RUN EXAPLES

In the case of our problem, constraints of arrangement can be divided into two categories: (1) Geometric constraints related to the physical placement of 3D objects. For example, the constraint of no_overlapping, constraint minimal_distance (the concerned object is away from other objects at least distance *dmin*). (2) Semantic constraints: from the fact that each object has a list of semantic attributes (light, temperature, vibration), this type of constraints uses these attributes to define the location of objects. For example, the temperature constraint uses the attribute temperature of the concerned object to place it away from sources of heat. It should be noted that several geometric constraints have been implemented allowing a first validation of our approaches. The implementation of semantic constraints is underway. The next part will be devoted to describe the firsts results obtained in order to validate our constraints implementation. It must be noted that we use the same propagation techniques for all the constraints. For each one, we use the same heuristics to select variables and their associated values. In this paper we present only one constraint: the minimal_distance.

Minimum-distance-constraint

This constraint forces involved objects (cows in this example) to be far-off by a distance greater than or equal to a distance (*dmin*) specified by the user. In addition this constraint can be useful for example to put an object away from sources of heats. As shown in the following figure, objects can be placed in the space while keeping a minimum distance of *dmin*.

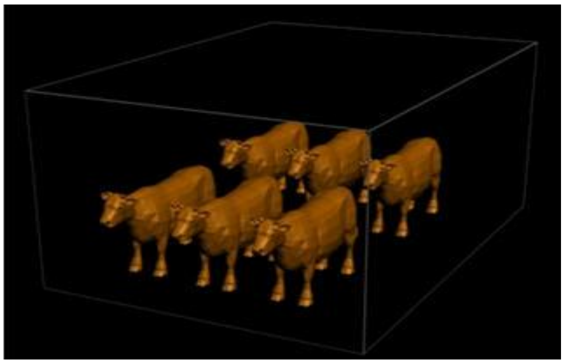

Figure 4. Illustration of the Minimum_distance_constraint

6.1 Response time

The response time is determined by the speed at which solutions are computed. Certain operations like displacing some expensive 3D objects and loading some big ones could slow down the 3D layout manipulations. So, interaction time cycle is depend not only on using the suitable technologies to model and implement constraint but also on the overall system architecture and interaction framework.

In order to evaluate the response time of our system, we have carry out some experiments with different number of objects and using only two kinds of constraints: minimum-distance and on_floor constraints. The following figure shows the response time as a function of the number of 3D objects.

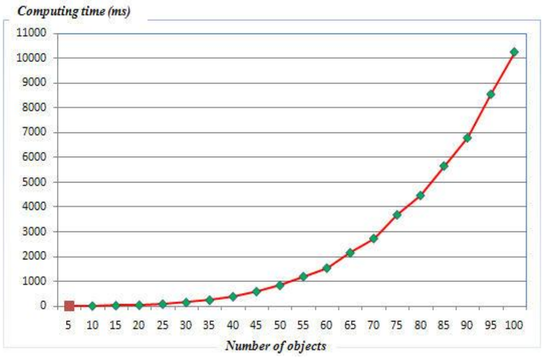

Figure 5. Computing time=f(number of objects)

6.2 Discussion

We can remark that the execution time increases with the number of objects to layout. Computing times obtained are sufficient to solve a real layout problem where the number of objects does not generally exceed fifty objects. Our current system can solve such problems in less than one second, which ensures real-time interaction

and a response almost instantly.

7. CONCLUSION

We have presented an efficient system for interactive 3D objects layout problem solving. Based on a combination RV-CP, this system can be used to assist the user during a 3D configuration task (spatial configuration). Through a GUI, the user can add 3D objects to the environment and choose constraints for each. 3D manipulations and user interaction will be converted to new queries sent, through a structured communication module, to the constraint solver. Thus each moving objects retriggers the solver which, after propagations of constraints, is looking for new solutions and transmit it to the VE. In the future, some tracks are envisaged: increase realism and immersion through the use of advanced 3D interaction techniques, increasing the size of the problem (increasing the number of objects to arrange).

REFERENCES

[Bowman, 1999] Bowman, D. (1999). Interaction Techniques for Common Tasks in Immersive Virtual Environments: Design, Evaluation, and Application. PhD thesis, Georgia Institute of Technology.

[Calderon et al., 2003] Calderon, C., Cavazza, M., and Diaz, D. (2003). A new approach to the interactive resolution of configuration problems in virtual environments. Lecture notes in computer science, 2733 :112 – 122.

[Diaz and Codognet, 2001] Diaz, D. and Codognet, P. (2001). Design and implementation of the gnu prolog system. Journal of Functional and Logic Programming, Vol. 2001, No 6.

[Kefi et al., 2010] Kefi, M., Richard, P., and Barichard, V. (2010). Interactive configuration of restricted spaces using virtula reality and constraints programming techniques. In International Conference on Computer Graphics Theory and Applications.

[Richard et al., 2006] Richard, P., Chamaret, D., Inglese, F., Lucidarme, P., and Ferrier, J. (2006). Human-scale virtual environment for product design : Effect of sensory substitution. The International Journal of Virtual Reality.

[Sanchez et al., 2002] Sanchez, S., Roux, O. L., Inglese, F., Luga, H., and Gaildart, V. (2002). Constraint-based 3dobject layout using a genetic algorithm.

[Schulte, 1997] Schulte, C. (1997). Oz explorer : A visual constraint programming tool. Proceedings of the Fourteenth International Conference on Logic Programming, pages 286–300.

[Xu et al., 2002] Xu, K., Stewart, J., and Fiume, E. (2002). Constraint-based automatic placement for scene composition. In Graphics Interface Proceedings, University of Calgary.

[Zorriassatine et al., 2003] Zorriassatine, F., Wykses, C., Parkin, R., and Gindy, N. (2003). A survey of virtual prototyping techniques for mechnanical product development. Journal of Engineering Manufacture, Part B :217.