Biologically-Plausible Reactive Behaviors for Robots and Virtual Characters

Abraham Sánchez L., Gerardo Garcia L. and René Zapata‡

Facultad de Ciencias de la Computación, BUAP
14 Sur esq. San Claudio, CP 72570
Puebla, Pue., México
‡ Robotics Department, University of Montpellier II
161 rue Ada, 34392 Montpellier Cedex 5 - France

Abstract. Reactive systems are a relatively recent development in robotics that has redirected artificial intelligence research. This new approach grew out of a dissatisfaction with existing methods for producing intelligent robotic response and a growing awareness of the importance of looking at biological systems as a basis for constructing intelligent behavior. This works addresses the modeling of the reactive behaviors for robots and virtual characters moving in unstructured and dynamic environments. This model is based on the definition of the interaction component (deformable virtual zone) of an internal state of the robot (or the character) and leads to avoidance-oriented control laws. Experimental results show the effectiveness of the proposed approach here.

1 Introduction

Collision avoidance has been an active research topic in Robotics. In general there have been two directions in this area: motion planning and reactive behavior. In motion planning, a collision-free path is planned an the robot is instructed to follow the path [7], [3]. In the later approach, collision avoidance is built as a reactive behavior of the robot. When the robot is close to an obstacle, it produces a repulsive force which will push the robot away and thus avoid collision [4].

Both approaches have advantages and disadvantages. The reactive behavior approach has a clear advantage in performance. In most cases it can be done in real time or near real time, even when the number of degrees of freedom (dofs) involved is large. Reactive methods can even avoid moving obstacles [10]. These methods suffer, however, from the local minima problem. While the motion planning approach does not suffer the local minima problem in general, it is usually too slow to be used in an interactive (changing) environment.

From path planning to trajectory control, the motion planning problem for robots has been thoroughly investigated in the case of structured environments. Moving among unknown or badly modeled environments, practically induces the necessity of taking unscheduled and dynamic events into account and reacting as the living beings would do. Therefore, reactive behaviors play a fundamental role when the robot has to move through unstructured and dynamic environments.

Advances in Computer Science and Engineering
Research in Computing Science 27, 2007, pp. 231-242

Received 09/02/07
Accepted 08/04/07
Final version 21/04/07

For the last fifteen years, the scientific community has been interested in the problem of reactive behaviors for collision avoidance in the domain of mobile robots [11]. This kind of algorithm is based on the definition of a protecting and deformable zone surrounding the robot (see next section).

In this trend, this work aims at providing a practical planner that considers reflex actions and PRM techniques to account for planning with changing obstacles. The paper is organized as follows. Section II gives an overview of the DVZ principle. Section III describes the reactive PRM approach for both holonomic and non-holonomic motion planning. A novel idea to solve the locomotion planning for virtual characters in dynamic environments is discussed in the section IV. Finally, the conclusions and future work are presented in section V.

2 Biologically-Plausible Reactive Behaviors

In many industrial, exploration or lab robotic tasks, a robot mainly has to perform some subtasks: (i) collision avoidance, (ii) target pursuit, (iii) localization, and (iv) path planning. The first three tasks obviously need perception capabilities, while the fourth one also does when the environment is subject to change (re-planning when a failure occurs).

The reactive control algorithm we use is based on the definition of a protecting and deformable zone surrounding the robot. The DVZ (Deformable Virtual Zone) is parameterized by the motion variables of the moving robot and can deform in the presence of distance information in the robot workspace. When an obstacle enters the sensor space, it induces a deformation of the DVZ that will be compensated by the robot motion controller. Therefore, the algorithm is a kind of 2-player game: the first one, i.e. the environment, induces undesired deformations; the second one, i.e. the robot controller, tries to rebuild the DVZ. This algorithm was first described in [11] for many applications and tested for robots moving in 2 dimensions [10], flying robots or autonomous submarines and also mobile manipulators [2]. This algorithm which was initially designed for obstacle avoidance, has two main advantages. First, the environment does not need to be a priori known, and second, the controller can take into account other constraints such as target pursuit, altitude maintenance, course control and so on. Figure 1 illustrates this general principle that will be described in the next paragraphs.

2.1 Mathematical Basis

The general framework for formalizing this principle is the category $\mathcal{D}$ of topologically equivalent sets of the $(n-1)$-dimensional unitary sphere $S^{n-1}(0,1)$ in $\mathbb{R}^n$. An object A of this category is related to the unitary sphere through an homeomorphism (imbedding of the sphere):

$$\delta_A : S^{n-1}(0,1) \to A \subset \mathbb{R}^n \tag{1}$$

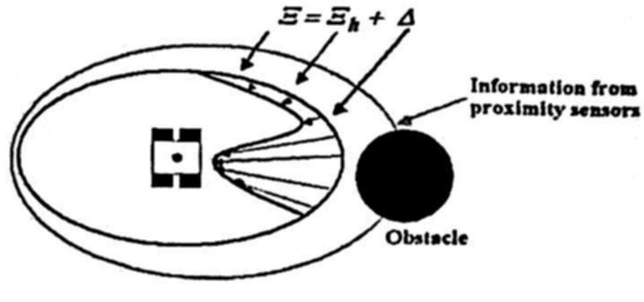

Fig. 1. DVZ principle in 2D.

The transformations between two objects A and B of $\mathcal{D}$ are the deformations obtained by the combination of the two defining homeomorphisms:

$$\delta_B \circ \delta_A^{-1} : A \to B \tag{2}$$

Let $R \subset \mathbb{R}^n$ be a convex rigid body, subset of the nD-space $\mathbb{R}^n$. The boundary ∂R of R can also be considered as the result of an imbedding of $S^{n-1}(0,1)$ in $\mathbb{R}^n$. We have $\partial R \in \mathcal{D}$ and:

$$\delta_R : S^{n-1}(0,1) \to \partial R \subset \mathbb{R}^n \tag{3}$$

Reciprocally, if f is an imbedding of ∂R in $\mathbb{R}^n$ with $E = f(\partial R)$, we can say that

$$f : \partial R \to E \subset \mathbb{R}^n \tag{4}$$

Any object $A \in \mathcal{D}$ separates $\mathbb{R}^n$ in two connected components, the interior $Int(A)$ of A and the exterior $Ext(A)$ of A. Therefore, we have $\mathbb{R}^n = Int(A) \oplus Ext(A) \oplus A$. A partial order is induced on $\mathcal{D}$ by the relation:

$$A \prec B \Leftrightarrow Int(A) \subset Int(B) \tag{5}$$

The rigid body R will represent a controlled robot moving among obstacles in $\mathbb{R}^n$. Any n-dimensional state vector characterizing the motion of R is denoted as a vector

$$\pi = [p_1\, p_2 \ldots p_n]^T \tag{6}$$

Axiom 1 *We assume that the robot R can be controlled by the derivative of this state vector. We note*

$$\phi = \dot{\pi} \tag{7}$$

Definition 1 *We define a DVZ of R as any imbedding Ξ of ∂R in $\mathbb{R}^n$ such that the relation $\partial R \prec \Xi(\partial R)$ holds. We have $\Xi(\partial R) \in \mathcal{D}$.*

Definition 2 *We define a controlled DVZ, Ξ_h, as a DV which depends on the state vector characterizing the motion of R:*

$$\Xi_h = \rho(\pi) \tag{8}$$

Definition 3 *Let $\mathcal{P} = (\Xi_h, \Xi)$ be a pair of two DVZ of R (the first one being a controlled DVZ) and such that $\Xi(\partial R) \prec \Xi_h(\partial R)$. We define the deformation $\triangle$ of the DVZ Ξ_h with respect to Ξ as the functional difference of Ξ and Ξ_h:*

$$\triangle = \Xi - \Xi_h \tag{9}$$

According to this definition, the deformation $\triangle$ is a one-one map that associates the vector $P - P_h$ to the point $M \in \partial R$, where $P = \Xi(M)$ and $P_h = \Xi_h(M)$. It can therefore be considered as a vector field defined on ∂R.

Axiom 2 *We assume that the robot can perceive distances in all directions of space. We also assume that the set of maximum distances that can be perceived by R and the set of actually perceived distance are two objects of the category $\mathcal{D}$, respectively named sensor boundary and information boundary (respectively denoted by Θ and Ψ), such that $\Psi \prec \Theta$. The deformation I of Θ with respect to Ψ is given by:*

$$I = \Psi - \Theta \tag{10}$$

The deformation I can also be considered as a vector field on ∂R.

Definition 4 *We define an uncontrolled DVZ, Ξ, as a DVZ which depends onf the sensor boundary deformation I:*

$$\Xi = \beta(I) \tag{11}$$

Let $\mathcal{P} = (\Xi_h, \Xi)$ be a pair composed of a controlled DVZ and an uncontrolled DVZ, the deformation $\triangle$ of the DVZ Ξ_h with respect to Ξ can be written:

$$\triangle = \Xi - \Xi_h = \beta(I) - \rho(\pi) \tag{12}$$

For a given point $M \in \partial R$, the deformation vector $\triangle(M)$ depends on the intrusion of proximity information $I(M)$, in the rigid body workspace and on the controlled DVZ Ξ_h.

By differentiating equation (12) with respect to time, we get:

$$\dot{\triangle} = -\nabla_\pi[\rho]\phi + \nabla_I[\beta]\psi \tag{13}$$

where ∇_ξ is the derivation operator with respect to the vectorial variable ξ and $\psi = \dot{I}$.

This equation can be rewritten as:

$$\dot{\triangle} = A\phi + B\psi \tag{14}$$

Variations in Δ are controlled by a 2-fold input vector $u = [\phi\ \psi]^T$. The first control vector ϕ, due to the robot controller tends to minimize deformation of the DVZ. The second one, ψ is unknown and induced by the environment itself (and could, at most, try to maximize these deformations). This equation will be referred as the main equation of the problem.

Once the main equation obtained, its integration (i.e., the obtention of a "good" control vector ϕ) can be computed in 2 steps:

1. Choosing the desired variation of this deformation as a function of the real deformation and its derivative:
$$\dot{\Delta}_{des} = -K_{prop}\Delta - K_{der}\dot{\Delta} \tag{15}$$
where K_{prop} and K_{des} are heuristically chosen.
2. Computing the best control vector $\check{\phi}$ at time t obtained by inverting equation (14) after replacing the deformation derivative by its desired value $\dot{\Delta}_{des}$:
$$\check{\phi} = A^{\dagger}(\dot{\Delta}_{des} - B\hat{\psi}) \tag{16}$$
where $A^{\dagger}$ is the inverse function (pseudo-inverse) of the linear function A and $B\hat{\psi}$ is an estimation of the second control vector ψ at time t obtained at time $t-1$:
$$B\hat{\psi}(t) = \dot{\Delta}_{measured}(t-1) - A\phi(t-1) \tag{17}$$

The control law (equation 16) tends to minimize the function $\|\dot{\Delta}_{des} - \dot{\Delta}\|$ in the least-squares sense. The ∞-dimensional functional equation 14 cannot, of course, be used directly. It is necessary to sample the sensor space in order to obtain an n-dimensional definition of the DVZ. This can be done by considering that the information vector has n dimensions (as many as the number of distance sensors). Equation 14 keeps its general form but all its entries are now matrices or vectors. Figure 2 shows some DVZ examples in 3D.

Fig. 2. Some DVZ examples in 3D.

In summary, collision avoidance is a 4-step process:

1. Measurement of the intrusion of information I as an n-dimensional vector (as many dimensions as the number of sensors).

2. Derivation of the deformation $\triangle$ and of its derivative $\dot{\triangle}$.
3. Estimation of the uncontrolled control vector $\hat{\psi}$.
4. Computation of the best control vector $\breve{\phi}$.

3 Reactive Behaviors in Motion Planning

Robot motion planning has led to active research over the two last decades. In particular, probabilistic techniques have received a lot of attention in recent years [3]. They have proven to be effective methods that can be applied to different problems arising in fields as diverse as robotics, graphics animation, computational biology.

Probabilistic roadmap method (PRM) is a general planning scheme building probabilistic roadmaps by randomly selecting configurations from the free configuration space and interconnecting certain pairs by simple feasible paths. The method has been applied to a wide variety of robot motion planning problems with remarkable success. PRM planners have been originally designed for solving multiple-query or single-query problems.

Dynamic changes in the environment are very common in many motion planning applications such as planning for evolving industrial environments, navigation in real or in virtual worlds.

The adaptation of PRM planners to environments with both static and moving obstacles has been limited so far. This is mainly because the cost of reflecting dynamic changes into the roadmap during the queries is very high. On the other hand, single-query variants, which compute a new data structure for each query, deal more efficiently with highly changing environments. They however do not keep the information reflecting the constraints imposed by the static part of the environment useful to speed up subsequent queries.

The proposed approach integrates the lazy PRM planning method [1], [6] and the reactive control by DVZ [11] in the following way: a collision-free feasible path for the robot is calculated by the lazy PRM method, the robot starts moving (under the permanent protection of its DVZ), in the absence of dynamic obstacles, the control is performed by the lazy PRM method and does not require reflex commands. If there are dynamic obstacles in its path, the reactive method takes the control and generates commands to force the robot to move away from the intruder obstacles and gives back its DVZ to the original state.

In this point, the robot has lost its original path, and it is necessary to search for a reconnection path to reach its goal. The new path found is a single collision-free curve of Reeds & Shepp for non-holonomic robots or a straight-line path for holonomic ones. If the attempt of reconnection is successful, the robot executes its new path towards the goal. The new alternative path was obtained with the lazy PRM method by using the information stored in the current robot's configuration, but if a deformation appears, the processes are interrupted by reflex actions that forces the planner to go back to the previous state.

After a successful reflex action, the mobile robot recovers the intact state of its DVZ, but its initial planned path will be lost. The lazy PRM method needs to have a path to push the robot to the goal and it will be necessary to provide a path for such aim. Due to the high computational cost of a complete replanning, the method will avoid it by executing a process that uses a single path to reconnect with the planned path.

If the reconnection attempts fails, it may happen that paths are blocked by many dynamic objects, or a moving object is parked obstructing the planned path. In this case, the planner executes the lazy PRM method (the initial configuration is the current configuration in the robot). The lazy PRM will be called several times until it returns a collision-free path. If after some attempts a collision-free path can not be found, the planner reports failure.

The algorithm can finish in three forms: i) the robot executes its path successfully, ii) the reflex action is not sufficient and a collision occurs, or iii) the robot does not find an alternative path to conclude its task.

In order to evaluate the performance of the proposed approach, we present some experimental results for car-like robots. The moving obstacles have a square form and move at constant velocity in straight line. Whenever they collide with another object they assume a new random direction in their movement.

Figure 3 shows an environment composed of four static obstacles and several dynamic obstacles moving randomly at the same velocity than the mobile robot. Figure 4 shows an environment with narrow passages.

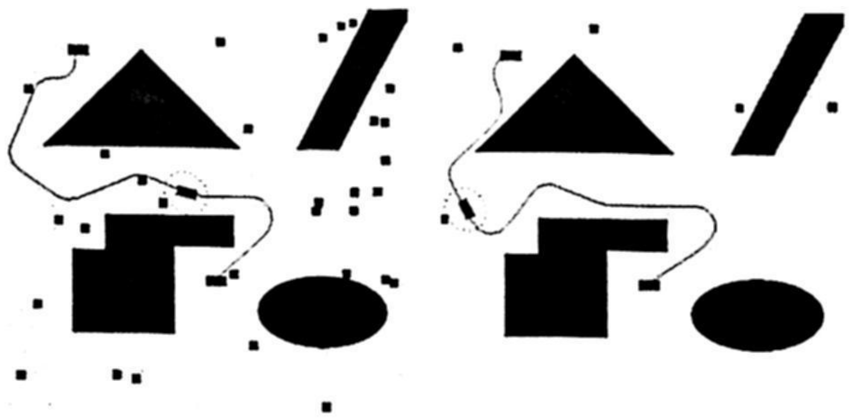

Fig. 3. An example of a query and its solution path (left) in an environment with 30 moving obstacles. The robot starts moving under the permanent protection of its DVZ (right), the scene contains 5 moving obstacles.

In fact, the method's performance can be considered satisfactory if it presents a fast planning phase, reflex actions based on sensors that do not require expensive algorithms, an effective process of reconnection performed in milliseconds, and a process of replanning that is executed if the Lazy PRM and DVZ' s parameters are appropriate. As mentioned in earlier sections, it can be considered that the methodology proposed here, includes these characteristics. The planning time is reduced due to the incomplete collision detector whose work is complemented with the robot's sensors during the path execution. On the other hand, the assignation of direction angles to the nodes that conform the shortest

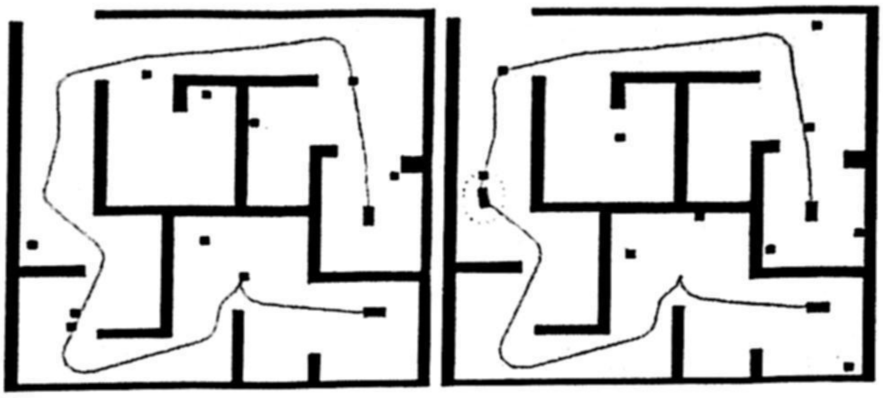

Fig. 4. An environment composed of some narrow passages. The scene contains 10 moving obstacles.

paths obtained by the algorithm A^*, produces curves that allow the algorithm to omit the optimization process (i.e., the smoothing process). With respect to the reconnection process, the paths obtained with the planner are conformed by a single Reeds & Shepp curve and based on the incomplete collision detector, making short the time and close to optimal the curves obtained with the algorithm. Since the reflex actions are provided by the DVZ method, it is possible to interrupt the reconnection and replanning processes if necessary, without incurring in bigger problems.

The proposed method in this section is general, even though we presented the case of motion planning for car-like robots. The DVZ's form is different, for manipulator robots we can use a cylinder.

4 Animating Reactive Motions

The synthesis of realistic human motion is a challenging research problem with broad applications in movies, cartoons, virtual environments, and games.

Due to the quality and realism of the result, the use of motion captured data has become a popular and an effective means of animating human figures [8]. However, since it is an inherently off-line process, there has been great interest in developing algorithms that are suitable for interactive applications. Designing appropriate control schemes can be difficult and only a limited number of methods consider reactive motions due to the presence of applied external forces [12], [5].

We propose to solve the locomotion planning problem for virtual characters evolving in a dynamic environment using a novel technique. While the legs and the pelvis of the virtual character follows a planned path, the animation of the upper part of the body is updated for 3D collision avoidance purposes (this happens normally in absence of dynamic obstacles, see Figure 5). In the presence of dynamic obstacles in its path, the reactive locomotion control takes the control and generates commands to stop the character and to let pass the intruder obstacles before forcing its DVZ to the original state.

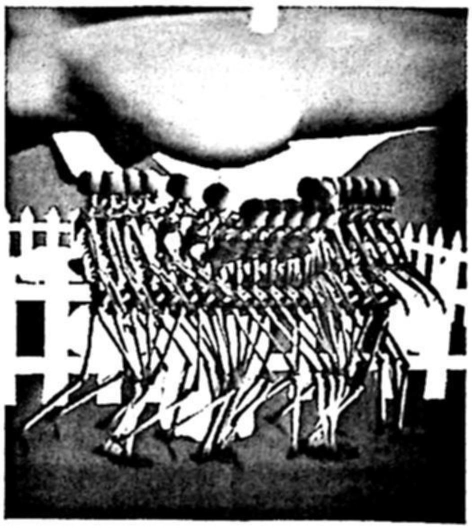

Fig. 5. Avoiding a branch.

Let us remember how it is possible to solve the collision avoidance problem for the reactive degrees of the character by using the warping module. The objective of the warping module is to locally modify the animation of the upper bodies of the character (arms and spine) when collisions occur in the animation produced by the locomotion controller. Each key-frame of the sequence is scanned and a collision test is performed. If a collision exists, the frame is marked. All the marked frames are gathered into connected subsequences, which are extended to create blocks absorbing collision-free frames in the neighborhood of the colliding subsequences. Such a subsequence extension is considered to provide smooth motions able to anticipate the corrective actions to be done. Each connected frame block is then processed independently. By following a similar idea to this proposal, it is possible to evaluate the deformation of the DVZ in each frame. Figure 6 illustrates our approach.

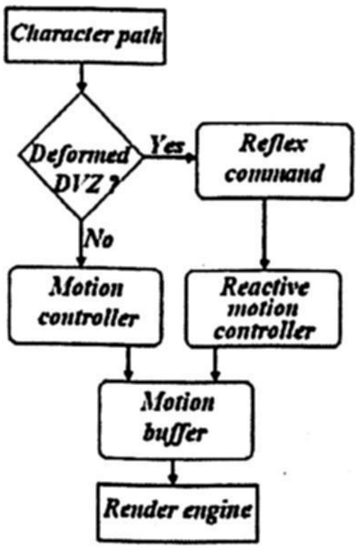

Fig. 6. An architecture for animation in dynamic environments.

In this work, we used a model of character of 52 degrees of freedom. The motion library is provided by the CMU Graphics Lab Motion Capture Database. Figure 7 illustrates the execution of the planner in a complex environment, a mall. The figures were trimmed, to show the part where the character walks and finds dynamic obstacles. The reflex commands in our approach are simples, because when the character finds dynamic obstacles, the most obvious and natural form to react is stopping the action (walk or running).

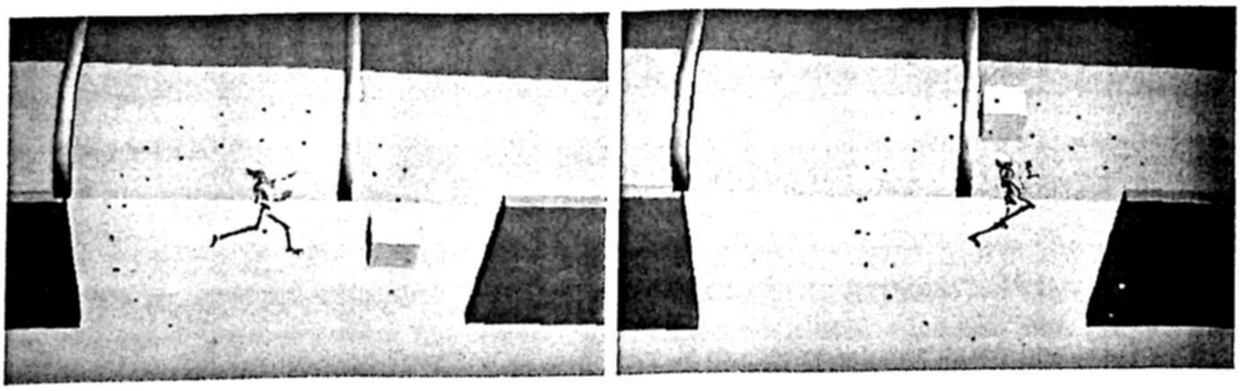

Fig. 7. Snapshots showing the action of the reactive controller to avoid dynamic obstacles.

The automated synthesis of motion for characters in unstructured environments is difficult because it requires solving a planning problem subject to multiple constraints. Obstacles in the environment constrain the motion in an obvious fashion, as typified by narrow passages. Other types of constraints include a character's joint limits, the requirements for balance, the character's natural disposition for particular postures and motion, and so on.

In this work, we have presented a novel idea to solve the locomotion planning problem for virtual characters in dynamic environments. This approach provides good results and mainly the running times are acceptable. Obviously, the free parameters of the DVZ are adjusted by hand, but they do not present greater problems. Also, we made tests in situations where the character must avoid dynamic obstacles and at the same time it must avoid collisions with static obstacles (when the upper part of the body encounters them) (see Figure 8).

5 Conclusions and Future Work

In recent years, robotics has been subject to promising advances in sensor and actuators hardware, sensory processing techniques and low-level control methods. Yet, the are has not been benefited to the full amount from the availability of powerful knowledge presentation tools and action calculi.

Many of the designers of reactive systems look to biology as a source of models for use in robots. Although the diversity of these efforts is significant, ranging from traditionally engineered systems to those that dedicate themselves

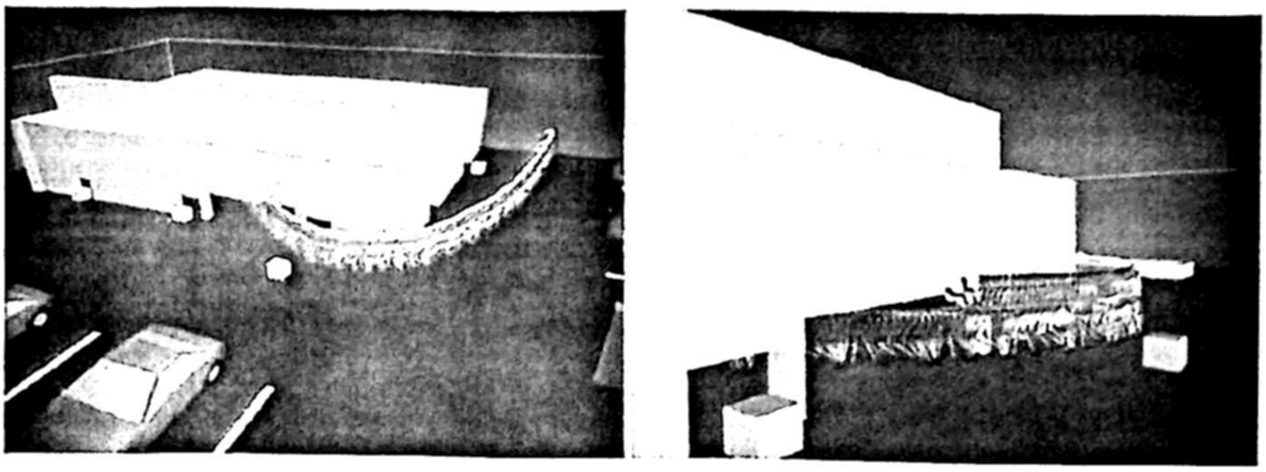

Fig. 8. Avoiding dynamic and static obstacles.

to faithfully replication biological behavior, this work reports on two examples that have affected reactive and hybrid system design.

The algorithms implementing the DVZ method can be seen as low-level interactions with the robot environment and must be coupled with high-level motion planning procedures. We have implicitly assumed that these high levels of control were existing to let the robot come back to its initial mission after a purely "reactive behavior".

A reactive lazy PRM planner for dynamically changing environments is presented in this paper. The results obtained in the evaluation of the reactive lazy PRM planner proposed in this work, show the importance of finding a solution for the complex problem of motion planning in dynamic environments.

Although some promising results are shown in its present form, the planner can be improved in a number of important ways. This approach can be extended to use real robots and to solve the problem posed by small static obstacles. Besides, some cases where the reflex actions are not sufficient to avoid collisions, were observed during the evaluation tests. Theses cases are difficult because they require a more intelligent behavior in order to avoid the robot to be trapped. In those cases, it can be necessary to add a process that computes the trajectories of moving objects and corrects the robot's path in real time.

Interactive generation of reactive motions for virtual humans as they are hit, pushed and pulled are very important to many applications, such as computer games, movies, cartoons, virtual environments. The use of motion captured data has become a popular mean of animating virtual characters, but since it is an off-line process, there has been great interest in developing algorithms that are suitable for interactive applications.

In this work, we have presented a novel idea to solve the locomotion planning problem for virtual characters in dynamic environments. This approach provides good results and mainly the running times are acceptable. Obviously, the free parameters of the DVZ are adjusted by hand, but they do not present greater problems. Also, we made tests in situations where the character must avoid dynamic obstacles and at the same time it must avoid collisions with static obstacles (when the upper part of the body encounters them).

The DVZ method can be used obviously with vision (instead of using other sensors like the telemetric or laser). For the collision avoidance scheme, the main input is the distance field in the robot front space. Stereovision allows this 3D reconstruction by measuring the disparity field in two images. By capturing images at successive step times, two or more cameras can also be used to detect motion in the robot workspace. Assuming that almost all points in the robot workspace are static, it is also possible to derive an expression of the ego-motion (self motion of the robot) and to fuse it with proprioceptive information.

References

1. Bohlin R. and Kavraki L. E. "Path planning using lazy PRM", *Proc. of the IEEE Robotics and Automation Conference*, (2000) 521-528
2. A. Cacitti, R. Zapata. Reactive behaviours of mobile manipulator based on the DVZ method, *Proc. of the IEEE Int. Conf. on Robotics and Automation*, (2001) 680-685
3. Kavraki L. E., Švetska P., Latombe J. C., and Overmars M. H. Probabilistic roadmaps for path planning in high-dimensional configuration spaces, *IEEE Transactions on Robotics and Automation*, Vol. 12, No. 4, (1996) 566-580
4. O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots, *International Journal of Robotic Research*, Vol. 5, No. 1, (1986) 90-98
5. T. Komura, H. Leung and J. Kuffner. Animating reactive motions for biped locomotion, *ACM VRST 04*, (2004)
6. Lanzoni C., Sánchez L. A., and Zapata, R. "A single-query motion planner for car-like nonholonomic mobile robots", *3rd International Symposium on Robotics and Automation (ISRA)*, (2002) 267-274
7. J. C. Latombe. Robot motion planning, *Kluwer Academic Publishers*(1991)
8. A. Sánchez L., J. Sánchez T. and R. Zapata, Planning motions for animated characters, *Journal Research on Computing Science*, Vo. 17, (2005) 265-274
9. J. Sánchez T., A. Sánchez L., M. Rodriguez C., X. Hernández V. and G. Garcia L., RMP3D: A multipurpose platform for motion planning, *Journal Research in Computing Science*, Vol. 20, (2006) 167-178
10. A. Sánchez L., R. Cuautle P., M. Osorio L. and R. Zapata. A reactive lazy PRM approach for non-holonomic motion planning, *J. S. Sichman et al. (Eds.): IBERAMIA-SBIA 2006, LNAI 4140*, (2006) 542-551
11. R. Zapata, P. Lépinay and P. Thompson. Reactive behaviors of fast mobile robots, *Journal of Robotic Systems*, Vol. 11, No. 1, (1994) 13-20
12. V. Zordan and J. Hodgins. Motion capture-driven simulations that hit and react, *ACM Computer Animation*, (2002)