IEEE Computer Animation 2001, H.-S. Ko (Ed.),
pp. 74–83, Seoul, Korea, November 2001

Through-the-eyes control of a virtual humanoïd

Nicolas Courty [*],
FRANCE TELECOM R&D Rennes
4 rue du Clos Courtel BP59
F-35012 Cesson Sevigne Cedex, France
Email Nicolas.Courty@irisa.fr

Éric Marchand [†], Bruno Arnaldi [‡]
IRISA - INRIA Rennes
Campus de Beaulieu,
35042 Rennes Cedex, France
Email Eric.Marchand@irisa.fr

Abstract

We present in this paper an animation technique to control a humanoïd in a virtual environment. The automatic generation of humanoïd motions is a difficult problem and there is a need to simplify it. The solution proposed in this paper consists in controlling the humanoïd motions through the image it "perceives": through-the-eyes control. *The considered approach is based on the visual servoing concept. It allows the automatic generation of (virtual) camera motions by simply specifying the task in the image space. This approach is suited to highly reactive contexts (video games, virtual reality). We also discuss of the integration of such techniques in a more complex behavioral system.*

1 Introduction

One of the most difficult issue in virtual reality consists in bringing more natural behaviors to the virtual characters of the inhabited virtual worlds. To gain reality and credibility, these autonomous actors have to perform complex computations in order to realize autonomous actions. Most of the time, the autonomous actor behavior follows the perception-decision-action steps. Perception can be achieved through a direct access to the world data-base or a subset of this data-base (e.g., [19, 21]) or through synthetic vision (e.g. [2, 8, 20, 11]). Once data related to the environment are collected, the decision module provides the actor with a set of actions to be executed. The animation of the avatar may be handled considering various techniques from full kinematic approaches to motion warping and other. These three modules have to be integrated in a more complex system that provides a task to be achieved by the actor.

In this paper, we will consider the case where the task given as input to the autonomous actor may be specified as a visual task. Though every task cannot be specified this way, most of interesting task for a humanoïd can be taken into account: gazing at an object, following moving objects or trajectory, tracking, etc. Considering this context, we will not envisage a perception-decision-action cycle. We proposed a complete framework that allows to directly control the humanoïd kinematic chain considering only visual informations, bringing a close link between the perception and action steps. The decision step is then in charge of giving visual tasks to be achieved. This approach is based on the visual servoing framework. This method has been first introduced in the robotics field [10, 6] and considered more recently in the computer animation context [7, 13]. Along with realization of the specified task, this approach allows the introduction of constraints in, for example, the joint trajectory. It is then possible to simply handle the joint limits problem that is fundamental in the humanoïd animation problem.

Overview of the paper The first section of this paper describes our motivation for considering an image-based animation system when dealing with visual perception within an autonomous actor. The second section introduces the visual servoing approach: how to control a multi-joint robot by vision and how to control its kinematic through visual servoing. We also discuss the enforcing of joint limits problem. The next section deals with the specification of the visual tasks used by the system, and the last section is dedicated to results.

2 Humanoïd animation

As far as we want the autonomous agent to perform complex behaviors in dynamic environments, we also require an appropriate simplification of the input parameters of the animation system. This abstraction for the control of the

[*] IRISA/INRIA Rennes - France Telecom R&D
[†] IRISA/INRIA Rennes (Vista project)
[‡] IRISA/INSA Rennes (Siames project)

humanoïd is needed [23]. The main idea is to be able to control directly the kinematic of an humanoïd by specifying visual tasks. This new technique may be one of the tools present in a toolbox for high level control of a virtual character. This vision-based abstraction of the control handles many requirements raised by classical behavioral models and avatar animation.

Behavioral requirements. The main drawback of classical behavioral techniques is that there is few interactions between perception (synthetic vision or database access) and the animation of the avatar body: in most cases the avatar "sees" its environment, then acts. In this case we may talk of passive perception, or also "guided" perception. This strong sequential process is far different from the way human beings act: the perception process is closely related to our movements. This is related to active or "autonomous" perception. Thus it is possible to distinguish three modes of perception [1] : one based on will (*endogenous*), one based on reactions (*exogenous*, for instance one object enters our perception field and draws our attention), and one passive (*idling*), our glance wanders around the scene. It is therefore clear that visual perception is driven by higher level modules (controlling the behavioral and cognitive aspects of the humanoïd) and commands to the kinematic of the humanoïd. Consequently, it is necessary to have, within the animation problematic, an animation system that can be reactive to changes in the environment, executing in real time, and allowing simple specifications of the tasks in the image space of what the humanoïd "sees". This problem is closely related to the works carried on in robotics, where robots can only perform tasks according to the information perceived from their cameras.

Autonomous actor representation. When dealing with the simulation of human's visual perception, one can wonder about the animation of the body with respect to it. There are many parts of the body that are concerned and involved in the perception process. Obviously, the eyes play an important part: they orientate the glance. Next, the neck, the torso and the remainder of the upper part of the humanoïd's body are involved in the perception process (as well as all the vertebraes of the spine). Moreover, when considering visual perception from a task level point of view, we might consider that locomotion may be directed by visual task: for example, if one wants to see a distant object in a close way, we have to generate a motion toward this object, for instance by connecting a locomotion module to the perception process. Hence there's a lot of different kinds of articulated chains that can be considered. In Fig 1, we present respectively two kinds of those chains that could be involved in the animation process: figure (a) contains nine degrees of freedom (three for the abdomen, one for the spine, three for the neck and two for the eyes). Figure (b) is a simplified model of the first one. The articulations are chosen arbitrary, but we'll see that the choice of those chains is related to the specification of the visual tasks to be achieved. The choice of the structural model of the skeleton can be much more complex, but at the cost of a bigger computing time. This can be seen as a criterion for animation level of details. The different kinds of kinematic chains require that the control system does not take as inputs the parameters in relation with those chains (e.g. joint values). By giving the humanoïd visual tasks, the abstraction of control may be performed.

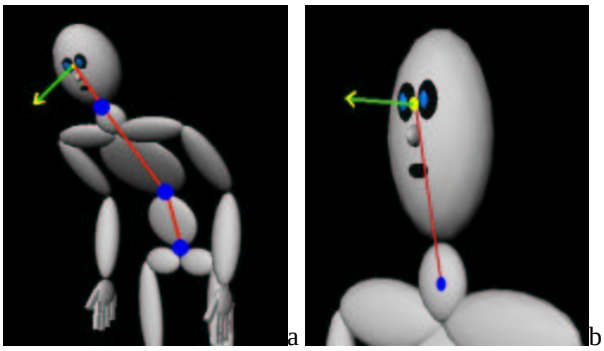

a b

Figure 1. Different types of articular chains when dealing with perception (a) nine degrees of freedom (b) five degrees of freedom

Animation requirements. We find in the literature many ways to animate a humanoïd body. One can consider physically realistic parameters, such as dynamic or optimal control. However such techniques are not easy to consider since their input parameters are difficult to handle, due to non-intuitive representations (muscles, torques,...). Some techniques use captured data (motion capture) and modify them to produce the desired motion with space constraints techniques [17], but it turns out that warping motions are not enough for automatic generation of new motions in dynamic environments. Using inverse kinematics control has also some drawbacks [15]: the given solution is local, and provides no guarantee to reach the goal. However, considering the redundancy, secondary tasks compatible with the main one can be used in an efficient way and provides flexibility of specification and intuitive control variables. In our case this solution seems the most adaptated, but the specification of the task remains in three dimensions, and may be hard to generate. Giving 2D constraints would be much more easier and intuitive to manipulate as already shown in [7, 13].

We choosed to link the visual information with the articular joints of the humanoïd by using a method similar

to inverse kinematic. The position/orientation of the eyes are given by the visual task, and is linked with the articular joints by the inverse of the Jacobian of the chain. Hence, the eyes constitute a "special" node of the inverse kinematic process. One of the main advantage of such a technique is to allow a closed loop control, which means that a new command is computed at each frame wrt. a visual constraint, allowing reactivity of the humanoïd to modifications of the environment.

Summary. These various requirements can be provided when using an image-based animation technique: the realism is obtained as far as the humanoïd is acting with respect to what it "perceives", and the specification are made easier in the sense that we only specify constraints in two dimensions, and no more in three dimensions. A possible architecture is depicted in figure 2. The animation system takes input from an animator's directives or behavioral engine, and generates adequate motions of the humanoïd body. The behavior engine may be represented by hierarchical finite states machines for instance, and can take inputs from an informed environment (the visual tasks can be contained into smart objects for instance). Those subjects are not in the scope of this paper (but are part of our animation system).

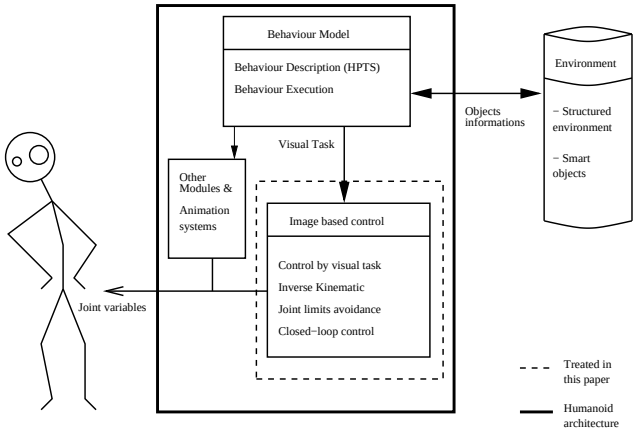


Figure 2. Architecture of the system

3 Image-based Humanoid control

Visual servoing has proved, within the robotics context, to be an efficient solution to these problems. Visual servoing or image-based camera control consists in specifying a task (mainly positioning or target tracking tasks) as the regulation in the image of a set of visual features [6, 9]. A set of constraints is defined in the image space (e.g., "I want to see the tree vertical and centered in the image while the head of the man must appear in the upper left part of the image"). A control law that minimizes the error between the current and desired positions of these visual features can then be automatically built. A good review and introduction to visual servoing can be found in [10]. As the task specification is carried out in 2D space, it does not require a 3D relationship between objects. However, since the approach is local, it is not *a priori* possible to consider planning issues. If the control law computes a motion that leads the camera to undesired configurations (such as occlusions, obstacles or joint limits), visual servoing fails. Control laws taking into account these "bad" configurations therefore have to be considered. Frameworks that allow the consideration of such constraints have been presented in, for example, [14, 13].

In [13] we presented the application of this framework to the definition within the computed animation problematic of the motion of a free camera. In this section we consider the case of a camera ending a constraint kinematic chains (i.e. humanoïd eye's). In a first step we link this camera to a poly-articulated chain standing for the body of the humanoïd. This introduces within the control problem the necessity to handle the joint limits problem. In the first part of this section we recall the general control issue and its application to visual servoing. We then propose an original and efficient solution to the enforcing of joint limits problem.

3.1 Visual servoing

General control issue. A general task can be seen as the regulation to zero of a task function [18] defined by [1]:

$$\mathbf{e} = \mathbf{J_1^+ e_1} + (\mathbf{I_n} - \mathbf{J_1^+ J_1})\mathbf{e_2} \tag{1}$$

where

- $\mathbf{e_1}$ is the main task to be achieved that induces m independent constraints on the n robot joints (with $m < n$).
- $\mathbf{e_2}$ is a secondary task.
- $\mathbf{J_1^+}$ and $\mathbf{I_n} - \mathbf{J_1^+ J_1}$ are two projection operators which guarantee that the camera motion due to the secondary task is compatible with the constraints involved by $\mathbf{e_1}$. $\mathbf{J_1} = \frac{\partial \mathbf{e_1}}{\partial \mathbf{q}}$ is the $m \times n$ full rank Jacobian matrix of task $\mathbf{e_1}$. Each column of $\mathbf{I_n} - \mathbf{J_1^+ J_1}$ belongs to Ker $\mathbf{J_1}$, which means that the realization of the secondary task will have no effect on the main task ($\mathbf{J_1}(\mathbf{I_n} - \mathbf{J_1^+ J_1})\mathbf{e_2} = 0, \forall \mathbf{e_2}$).

To make $\mathbf{e}$ decreases exponentially (i.e., $\dot{\mathbf{e}} = -\lambda \mathbf{e}$), we get:

$$\dot{\mathbf{q}} = -\lambda \mathbf{e} - (\mathbf{I_n} - \mathbf{J_1^+ J_1}) \frac{\partial \mathbf{e_2}}{\partial t} \tag{2}$$

where:

[1] If $\mathbf{M}$ is a matrix, we note $\mathbf{M}_{\bullet i}$ its i^{th} column and $\mathbf{M}_{i\bullet}$ its i^{th} row.

- $\dot{\mathbf{q}}$ is the joint velocity given as input to the robot controller;
- λ is the proportional coefficient involved in the exponential decrease of $\mathbf{e}$;

Vision-based tasks. When the task $\mathbf{e_1}$ to be achieved is a vision task, this one is specified as the regulation in the image of a set of visual features[6][9][10]. We can applied the previous control law to vision-based tasks. To define such a task, the animator will have to define the set of visual features and their desired positions in the image plane.

Let us denote $\mathbf{P}$ the set of selected visual features used in the visual servoing task. To ensure the convergence of $\mathbf{P}$ to its desired value $\mathbf{P_d}$, we need to know the interaction matrix (or image Jacobian) $\mathbf{L_P^T}$ that link the variation (the motion) $\dot{\mathbf{P}}$ of the visual feature in the image to the camera motion $\mathbf{T_c}$ [6] :

$$\dot{\mathbf{P}} = \mathbf{L_P^T T_c} \tag{3}$$

Control laws in visual servoing are generally expressed in the operational space (i.e., in the camera frame), and then computed in the articular space using the robot inverse Jacobian. However, in order to combine a visual servoing with the enforcing of joint limits, we have to directly express the control law in the articular space. Indeed, joint limits are defined in this space.

This leads to the definition of a new interaction matrix such that:

$$\dot{\mathbf{P}} = \mathbf{H_P}\ \dot{\mathbf{q}} \tag{4}$$

Since we have $\mathbf{T_c} = \mathbf{J(q)}\ \dot{\mathbf{q}}$, where $\mathbf{J(q)}$ is nothing but the robot Jacobian, we simply obtain:

$$\mathbf{H_P} = \mathbf{L_P^T}\ \mathbf{J(q)} \tag{5}$$

If l visual features are selected, the dimension of $\mathbf{H_P}$ is $l \times n$. If the visual features are independent, the rank m of $\mathbf{H_P}$ is equal to l, otherwise $l > m$. The vision-based task $\mathbf{e}_1$ is then defined by:

$$\mathbf{e_1} = \mathbf{C}(\mathbf{P} - \mathbf{P_d}) \tag{6}$$

where $\mathbf{P_d}$ is the desired value of the selected visual features, $\mathbf{P}$ is their current value (measured from the image at each iteration of the control law), and $\mathbf{C}$, called combination matrix, has to be chosen such that $\mathbf{J_1} = \mathbf{CH_P}$ is full rank along the desired trajectory $\mathbf{q_r}(t)$. It can be defined as $\mathbf{C} = \mathbf{W}{\mathbf{H}^+}_{\mathbf{P}|\mathbf{P}=\mathbf{P_d}}$, where $\mathbf{W}$ is a full rank $m \times n$ matrix such that Ker $\mathbf{W}$ = Ker $\mathbf{H_P}$ (see [18][6] for more details). If $\mathbf{H_P}$ is full rank m, we can set $\mathbf{W} = \mathbf{H_P}$, then $\mathbf{C} = \mathbf{I}$ and $\mathbf{J_1} = \mathbf{H_P}$ is a full rank $m \times n$ matrix. If rank m of $\mathbf{H_P}$ is less than l, we have $\mathbf{J_1} = \mathbf{W}\mathbf{H_P^+}\mathbf{H_P}$ which is also a full rank $m \times n$ matrix.

3.2 Enforcing of joint limits.

Issue and related work. Dealing with humanoïd control, joint limits avoidance is a fundamental process that has to be implemented to achieve realistic motion.

The most classical way to solve this problem is to define the secondary task as the gradient of a cost function h_s ($\mathbf{e_2} = \frac{\partial h_s}{\partial \mathbf{q}}^T$). This cost function must reach its maximal value near the joint limits and its gradient must be equal to zero when the cost function reaches its minimal value [18]. Several cost functions h_s which reflect this desired behavior have been presented in [18, 3, 12, 16, 22]. We briefly recall the most efficient of the cost functions proposed in [12].

Let us denote $\bar{\mathbf{q}}_{i_{min}}$ and $\bar{\mathbf{q}}_{i_{max}}$ the lower and upper limits that have not to be crossed (the joint limits). Activation thresholds on axis i are defined by $\tilde{\mathbf{q}}_{i_{min}}$ and $\tilde{\mathbf{q}}_{i_{max}}$ such that:

$$\begin{array}{l} \tilde{\mathbf{q}}_{i_{min}} = \bar{\mathbf{q}}_{i_{min}} + \rho\ \Delta\bar{\mathbf{q}} \\ \tilde{\mathbf{q}}_{i_{max}} = \bar{\mathbf{q}}_{i_{max}} - \rho\ \Delta\bar{\mathbf{q}} \end{array} \tag{7}$$

where $\Delta\bar{\mathbf{q}} = \bar{\mathbf{q}}_{i_{max}} - \bar{\mathbf{q}}_{i_{min}}$ and $0 < \rho < 1/2$ (typically, $\rho = 0.1$). This means that when an axe is in a critical area (i.e. located between $\tilde{\mathbf{q}}$ and $\bar{\mathbf{q}}$) a motion must be generated to move the axes in the opposite direction. A cost function that achieve this goal is thus given by:

$$h_s = \alpha \sum_{i=1}^{n} \frac{s_i^2}{\Delta\bar{\mathbf{q}}} \quad \text{where } s_i = \left\{ \begin{array}{ll} \mathbf{q}_i - \tilde{\mathbf{q}}_{i_{max}} & \text{if } \mathbf{q}_i > \tilde{\mathbf{q}}_{i_{max}} \\ \mathbf{q}_i - \tilde{\mathbf{q}}_{i_{min}} & \text{if } \mathbf{q}_i < \tilde{\mathbf{q}}_{i_{min}} \\ 0 & \text{else} \end{array} \right. \tag{8}$$

The parameter α that sets the amplitude of the control law due to the secondary task is very important (see equation (1)). Indeed, as pointed out in [3], if α is too small, $\mathbf{e_2}$ may be insufficient to avoid a joint limit. Furthermore, if α is too large, it will result in some overshoot in the effector velocity. Therefore α is usually set based on trial and errors. We now propose a simple new solution to this important problem.

A proposed approach The approach we proposed in order to avoid joint limits is quite simple [4]. A good solution to achieve the avoidance task is to cut any motions on axes that are in critical area or that move the robot toward it. Considering that $\mathbf{q}_k$ is one of these axes, we have to compute a velocity $\dot{\mathbf{q}}_k = 0$.

With respect to equation (1), a more general task function that uses redundancy can be defined by:

$$\mathbf{e} = \mathbf{J_1^+ e_1} + \sum_{i=1}^{n_\mathbf{a}} \mathbf{a}_i \mathbf{E}_{\bullet i} \tag{9}$$

where

- $n_\mathbf{a} = \dim \text{Ker}\mathbf{J_1} = n - m$.

- $\sum_{i=1}^{n_\mathbf{a}} \mathbf{a}_i \mathbf{E}_{\bullet i}$ defines motions that try to ensure that the robot will never encounter its joints limits. Within this term:
  - $\mathbf{E}$ is a basis of Ker $\mathbf{J_1}$ of dimension $n \times n_\mathbf{a}$. By this way, the computed motions will have no effect on the main task.
  - $\mathbf{a}$ is a vector of gains that will be automatically computed.

Consider that several axes are in critical situation, we determine vector $\mathbf{a}$ in order to stop the motion on these axes. From equation (9), for each axis k in critical situation we obtain:

$$\dot{\mathbf{q}}_k = 0 \quad \Longleftrightarrow \quad \sum_{i=1}^{n_\mathbf{a}} \mathbf{a}_i \mathbf{E}_{ki} = -\left(\mathbf{J_1^+ e_1}\right)_k \tag{10}$$

If $p_\mathbf{a}$ axes are in critical situation, we can define from equation (10) a linear system $\mathbf{Fa} = \mathbf{s}$ where $\mathbf{F}$ is of dimension $p_\mathbf{a} \times n_\mathbf{a}$ while $\mathbf{s}$ and $\mathbf{a}$ are of dimension $n_\mathbf{a}$. We have three possible cases:

- when $p_\mathbf{a} > n_\mathbf{a}$, we have more axes in critical situation than redundant axes. Of course in that case, the total efficiency of the method cannot be ensured.
- when $p_\mathbf{a} = n_\mathbf{a}$, there is only one solution but the problem can be solved.
- when $p_\mathbf{a} < n_\mathbf{a}$, the system features multiple solutions.

In any case, a solution is given by $\mathbf{a}^* = \mathbf{F}^+\mathbf{s}$. The resulting control law is given by:

$$\dot{\mathbf{q}} = -\lambda \left(\mathbf{J_1^+ e_1} + \sum_{i=1}^{n_\mathbf{a}} \mathbf{a^*}_i \mathbf{E}_{\bullet i} \right) \tag{11}$$

Produce a null motion $\dot{\mathbf{q}}_k = 0$ for axes in critical situation is not always required nor desirable. To produce a smooth decay of the axis velocity, we modify the linear systems (10) in order to weight, with a coefficient γ_k, the participation of an axis to the avoiding process function of its distance to the joint limit. Between the joint limit $\bar{\mathbf{q}}$ and a threshold $\tilde{\mathbf{q}}$, the velocity of the corresponding axis should decrease and must stop when it reaches $\bar{\mathbf{q}}$.

$$\sum_{i=1}^{n_\mathbf{a}} \mathbf{a}_i \mathbf{E}_{ki} = -\gamma_k \left(\mathbf{J_1^+ e_1}\right)_k \tag{12}$$

Improvements to this approach are presented in [4].

4 Applications

4.1 Defining Visual tasks

Positionning task One of the difficulties in image-based visual servoing is to derive the image Jacobian $\mathbf{L}^T$ which corresponds to the selected control features. A systematic method has been proposed to analytically derive the interaction matrix of a set of control features defined upon geometrical primitives [6]. Any kind of visual information can be considered within the same visual servoing task (coordinates of points, line orientation, surface or more generally inertial moments, distance, etc).

Knowing these interaction matrices, the construction of elementary visual servoing tasks is straightforward. A large library of elementary skills can be proposed. The current version of our system allows to define X-to-Y feature-based tasks (or constraints) with X and Y defined in {point, line, sphere, cylinder, circle, etc.}. Defining X-to-Y featured-based tasks means that the operator wants to see the object X "on" the object Y in the image space. Let us note that X and Y are not necessarily the same features: for example, a point-to-line constraint means that we want to see a 2D point on a 2D line. Using these elementary positioning skills, more complex tasks can be considered by "stacking" the elementary Jacobians. For example if we want to build a positioning task wrt. to a segment, defined by two points $\mathbf{P_1}$ and $\mathbf{P_2}$, the resulting interaction matrix will be defined by:

$$\mathbf{L}_\mathbf{P}^T = \begin{bmatrix} \mathbf{L}_{\mathbf{P_1}}^T \\ \mathbf{L}_{\mathbf{P_2}}^T \end{bmatrix} \tag{13}$$

where $\mathbf{L}_{\mathbf{P_i}}^T$ is defined, if $\mathbf{P_i} = (X, Y)$ and z is its depth, by (See [6] for its derivation):

$$\mathbf{L}_\mathbf{P}^T = \begin{pmatrix} -1/z & 0 & X/z & XY & -(1+X^2) & Y \\ 0 & -1/z & Y/z & 1+Y^2 & -XY & -X \end{pmatrix} \tag{14}$$

More positioning skills can thus be simply defined.

Using this method The inputs provided to the system by an animation system or an animator are then quite simple: definition of a virtual primitive, definition of a desired position in the image space, and linking this virtual primitive to a real object of the environment. For instance, if a virtual primitive (such as a sphere) is linked to a moving object, the parameters of the virtual sphere must be updated at each frame according to the motion and deformation of the object (as example, the virtual sphere can stand for the bounding sphere of the object). All these informations can be enclosed in the definition of the object at creation stage.

4.2 Connecting with locomotion

We can consider that the locomotion is part of the visual perception activity. For example one can get closer to an object to see it, or when following an object with the look one might have to move toward it's direction or toward a new point of view, while maintaining the focus task. They are two ways to handle locomotion in such a context: directed by an independant process, or directed by the visual servoing process.

Locomotion as extern module In this case the locomotion is directed by the behaviour model, which is in charge to guarantee the coherency between the visual task and the locomotion process. In this case, the visual servoing process "undergoes" the locomotion, it has to react properly to the motions of the humanoïd (wrt. his kinematic limits).

Visual servoing based locomotion In this case, the locomotion is directed by the visual servoing process. We add some translationnal degrees of freedom at the root of the articular chain (we assume that the root of the chain is then situated at the center of the humanoïd, just as in the HAnim norm). The control law takes into account those translations, and generate translationnal velocities to ensure the visual task. The locomotion module (which can consist in motion capture, inverse kinematic, etc.) takes input from a controller which converts those speeds into commands that are compatible with his inputs (in the example presented in 5.1 the locomotion process takes as inputs a speed and a direction).

As far as the real displacement of the humanoïd does not match exactly the translations generated by the control law, an error is introduced (the decrease of the error is no more exponential), but as far as we generate a new velocities at each iteration, we ensure convergence. The use of the secondary task can be used to solve non-trivial problems as path planning or path following, avoiding occultations [13]... Those subjects have been considered but will not be treated in this paper.

5 Results

The validation of the theoritical part was accomplished on three types of humanoïd: respectively five, nine and twelve degrees of freedom. For visualisation purpose we choosed to use the GASP (General Animation and Simulation Platform) framework [5], developped at SIAMES project. The geometrical models for the humanoïds were compatible with the HAnim stantard. All the animations were performed in real time, the computing time for the command law was of the order of the millisecond with an O2 Silicon computer. Thus, all the demos presented were running with a 30 frames/sec frequency.

5.1 Basic examples

Tracking a moving object. As a basic example we choose to servo on a point situated at the center of a sphere. The task for the humanoïd is to move the red ball in a predefined position. The task given as input of our animation system is to center the ball all along the animation. It is completely independant of the motion of the arm, and has not any knowledge about the future position of the ball. The only knowledge available about the scene is the position of the center of the ball at each iteration.The figure 3 contains results where the first strip of images is an overall view of the scene, and the second strip is an humanoïd point of view.

In figure 4, we present the evolution of the error in the image (along the two axes). During the first iterations, the error is decreasing exponentially. We can notice that when the humanoïd has grabbed the ball, and starts to move it, the error is no more null, due to a delay in the perception action cycle. This tracking error can be fully compensated by introducing a term in equation (1) that allows to consider the motion of the target (see [13] for more details).

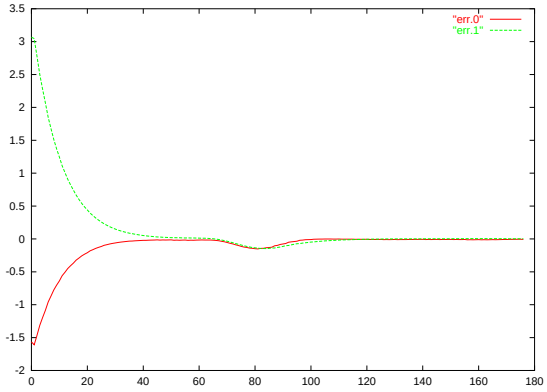


Figure 4. Evolution of the error (X-axis: frame, Y-axis: error value)

The painting example In this example, we consider a focusing task hard to handle with basic animation techniques. The goal of the visual task is to see a painting centered in the image. The specification of the task is done as following: four points matching with the corners of the painting are chosen. We choose their desired locations such as the painting appears centered in the image. The interaction matrix is then defined as the stacking of the four Jacobians relative to each of the four points. It is a $8 * 6$ matrix of

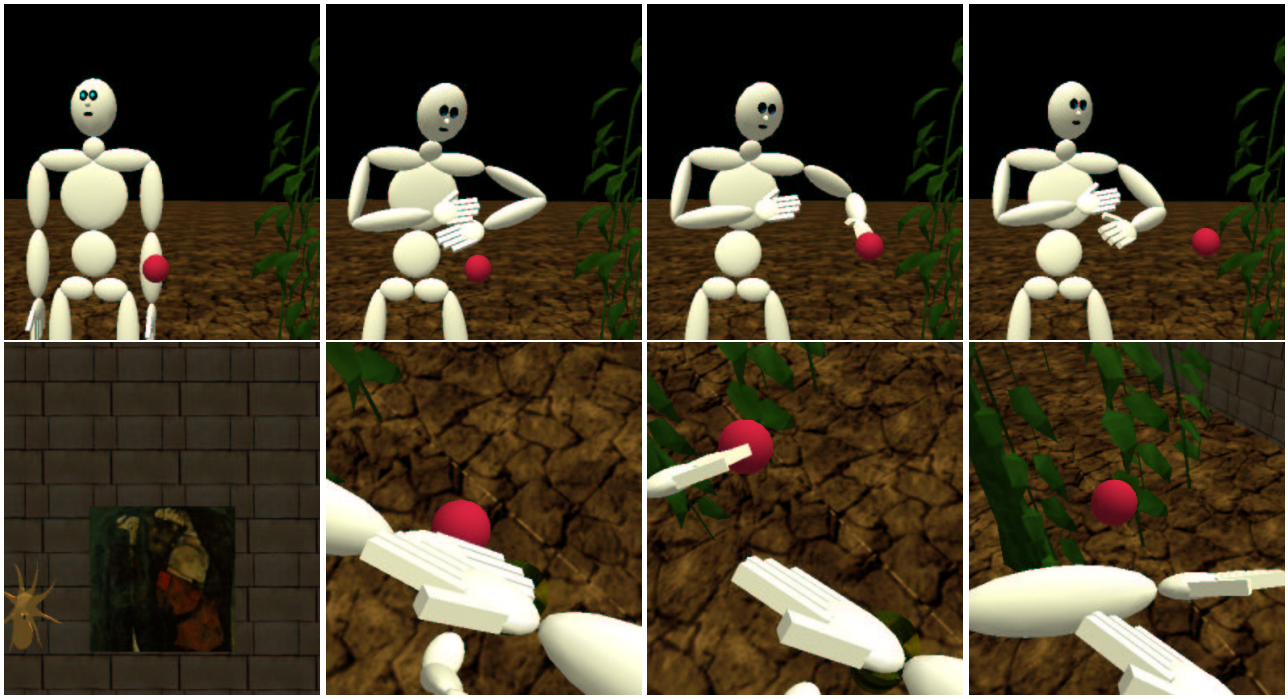
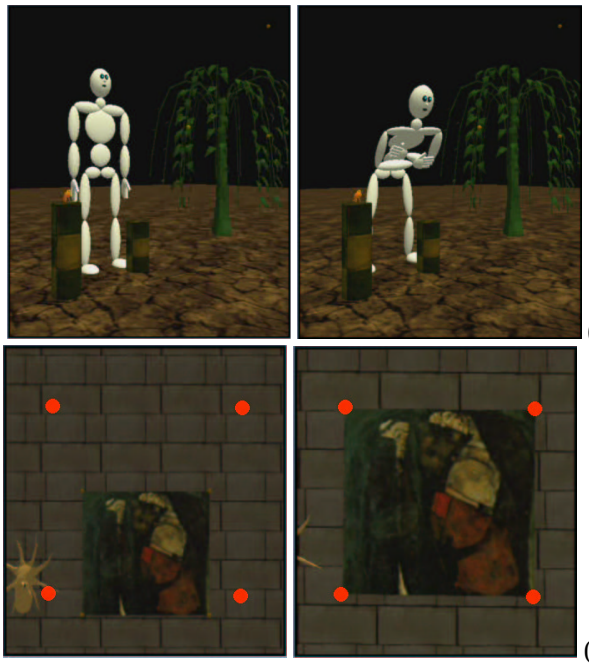


Figure 3. Gazing at a moving object

rank six, which creates a rigid link between the humanoïd's eyes and the painting. The resulting animation is presented in figure 5 (the painting is not in the field of view of figure 5(a)). In the images rendered from the humanoïd's point of view, we drawed the four desired positions of the points (in red). At the end of the animation, they match the four corners of the painting. Thus, the specification for this task has been done in two steps: specify the four corners of the painting, and place their desired positions in the image. From a behavioral point of view, we can assume that the desired position of the four corners may be an information contained in the painting (i.e., looking the painting would mean get the positions of the four corners).

Looking accurately a distant object This example is dedicated to the link between locomotion and the servoing process. In this case, the given visual task is to have a close look at a ball located in a distant place. To achieve such a task, we choose to servo on a virtual sphere, that has to be centered and at a distance z_d in the camera frame. This constrains the focusing task and the distance to the object (as far as we control the radius R of its projection in the image). In this case, the desired position in the image is given by $\mathbf{L}_{\mathbf{P}}^T = (0, 0, \pi R^2)$ where πR^2 is the desired surface of the circle in the image. The complete Jacobian for a centered sphere is given (see [6] for its complete derivation) by:



(a)

(b)

Figure 5. Servoing on four points: (a) extern view of the humanoïd (b) image "perceived"

$$\mathbf{L}_{\mathbf{P}}^T = \begin{pmatrix} -1/z_d & 0 & 0 & 0 & -1-R^2 & 0 \\ 0 & -1/z_d & 0 & 1+R^2 & 0 & 0 \\ 0 & 0 & \pi R^2/z_d & 0 & 0 & 0 \end{pmatrix} \quad (15)$$

We used a kinematic chain containing translational degrees of freedom. The results are presented in figure 6.

One difficulty in this example is that a simple solution for the humanoïd to get closer to the target is to lean over. This results in unnatural motions. A solution is to only consider, in a first time, a focusing task, and generate the motion toward the target with a secondary task (trying to minimize the distance between the humanoïd and the target). In a second time, we consider the complete task, when the humanoïd is close enough to the target.

5.2 A behavioral example

This example deals with the integration of this technique in a behavior. The aim of this animation is to exhort the simplicity of the specification of the parameters. The behavior described here is a model of passive perception. The goal for the humanoïd is to look at balls, and touch them. In our example, there are three balls present in the scene. We choosed to servo on spheres. The error in the image is computed for the three balls. As soon as this error is below a given threshold (which corresponds to the condition "the humanoïd is seing the ball"), a servoing task is executed. The basic motion for the humanoïd is a rest motion. While executing this motion, some of the balls appear in the perception field of the humanoïd, causing the steering behavior (i.e. the servoing) to be executed. While ensuring the servoing task, other balls appear in the perception field, ad so on. There is also a synchronization with a grasping motion: the humanoïd looks at the ball, after what it touchs it, then it swaps to another target. This behavior has been encoded within an hierarchical finite states machine. A ball entering the perception field provokes the "stacking" of a new target (which is the current position and radius of the sphere). The control law takes as input this data, and a predefined position in the image space (the ball must be seen centered in the image, with a suitable radius). The figure 7 shows the corresponding animation.

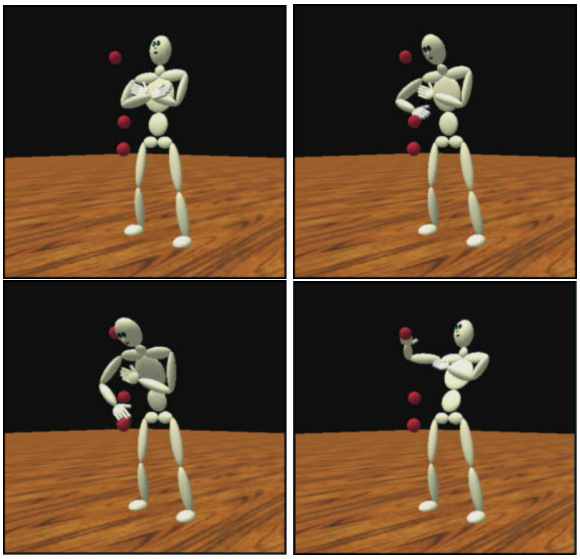


Figure 7. Passive perception as a behavior

6 Conclusions and future works

Controlling a humanoïd body in an unknown environment appears to be an awkward task. First, it is necessary to provide easy ways to animate the body of the humanoïd with respect to different tasks, and when dealing with autonomous agent a capacity to react dynamically to the modifications of the environment is required. The proposed method is dedicated to the simulation of visual perception for these agents. Our goal was to create an animation method with simple input parameters that can be easily connected to a more general behavioral model. We chose to use a technique widely used in the robotics community: visual servoing. This image-based control constitues the first novelty of our approach. The simplification of the input parameters is accomplished with a specification of the task in a *2D space*, while ensuring the good motions for the humanoïd in the *3D space*. Furthermore, the reactivy of the system is ensured with a closed loop control of the visual task. The main drawback of this technique comes up with the fact that we don't have a precise control of the trajectories in the articular space of the humanoïd. This could lead to unrealistic motions. Though we have introduced in this paper an efficient joint limit avoidance technique, it seems that it could be interesting to add physics parameters (as weight and gravity), and use the secondary task to constrain the motions (e.g. try to bring all the articulations in the less effort posture). Ongoing work considers the integration of this technique in a behavioral model. This integration generates new problems: sequencing the visual tasks, choosing these tasks appropriately wrt. particular behaviors (for instance steering, idling, search in an unknown environment) and planning for avoiding objects and occultations.

Acknowledgment. The authors wish to thank François Chaumette for is valuable comments, and Stéphane Ménardais for the locomotion module.

References

[1] N.I. Badler, D. Chi, S. Chopra. Virtual human animation based on movement observation and cognitive behaviour models. In IEEE *Computer Animation´99*, pages 128–137. Geneva, Switzerland, May 1999.

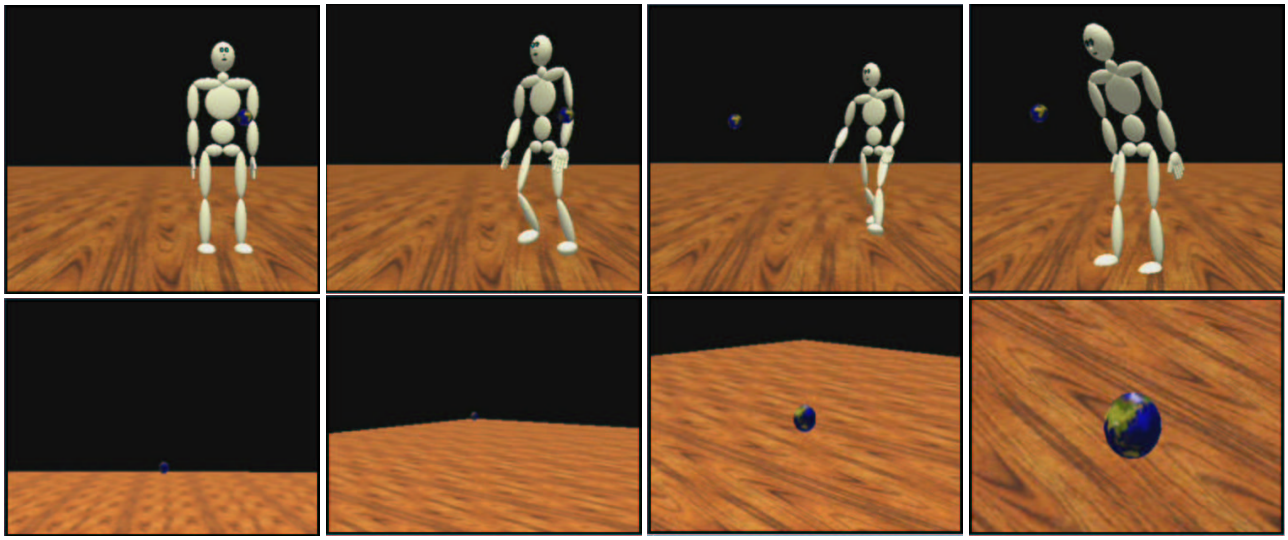


Figure 6. Connecting with locomotion

[2] B. Blumberg. Go with the flow: Synthetic vision for autonomous animated creatures. In *AAAI Conference on Autonomous Agents*, 1997.

[3] T.-F. Chang, R.-V. Dubey. A weighted least-norm solution based scheme for avoiding joints limits for redundant manipulators. *IEEE trans. on Robotics and Automation*, 11(2):286–292, April 1995.

[4] F. Chaumette, E. Marchand. A new redundancy-based iterative scheme for avoiding joint limits: Application to visual servoing. In *IEEE Int. Conf. on Robotics and Automation*, volume 2, pages 1720–1725, San Francisco, CA, April 2000.

[5] S. Donikian, A. Chauffaut, T. Duval, R. Kulpa. Gasp: from modular programming to distributed execution. In *Computer Animation'98*, Philadelphia, USA, June 1998.

[6] B. Espiau, F. Chaumette, P. Rives. A new approach to visual servoing in robotics. *IEEE trans. on Robotics and Automation*, 8(3):313–326, June 1992.

[7] M. Gleicher, A. Witkin. Through-the-lens camera control. In *Proc. of SIGGRAPH 92, in Computer Graphics Proceedings*, pages 331–340, July 1992.

[8] D.Thalmann H. Noser, O. Renault, N. Magnenat-Thalmann. Navigation for digital actors based on synthetic vision, memory and learning. *Computer & Graphics*, 19(1):7–19, 1995.

[9] K. Hashimoto. *Visual Servoing : Real Time Control of Robot Manipulators Based on Visual Sensory Feedback*. World Scientific Series in Robotics and Automated Systems, Vol 7, World Scientific Press, Singapor, 1993.

[10] S. Hutchinson, G. Hager, P. Corke. A tutorial on visual servo control. *IEEE trans. on Robotics and Automation*, 12(5):651–670, October 1996.

[11] J.J. Kuffner, J.C Latombe. fast synthetic vision, memory, and learning models for virtual humans. In IEEE *Computer Animation '99*, pages 118–127, Geneva, Switzerland, may 1999.

[12] E. Marchand, F. Chaumette, A. Rizzo. Using the task function approach to avoid robot joint limits and kinematic singularities in visual servoing. *Proc of IEEE/RSJ Int. Conf on Intelligent Robots and Systems, IROS'96*, 14(3):1083–1090, November 1996.

[13] E. Marchand, N. Courty. Image-based virtual camera motion strategies. In *Graphics Interface Conference, GI2000*, pages 69–76, Montréal, Quebec, May 2000.

[14] B. Nelson, P.K. Khosla. Integrating sensor placement and visual tracking strategies. In *IEEE Int. Conf. Robotics and Automation*, volume 2, pages 1351–1356, San Diego, May 1994.

[15] P. Baerlocher, R. Boulic. Task-priority formulations for the kinematic control of highly redundant articulated structures. In *Proc. of IROS'98*, Victoria, Canada, October 1998.

[16] C. Phillips, J. Zhao, N. I. Badler. Interactive real-time articulated figure manipulation using multiple kine-

matic constraints. In *ACM Symposium on Interactive 3D Graphics*, pages 245–250, March 1990.

[17] C. F. Rose, B. Guenter, B. Bodenheimer, M. F. Cohen. Efficient generation of motion transitions using space-time constraints. In *SIGGRAPH 96 Conference Proceedings*, pages 147–154, New Orleans, Louisiana, August 1996.

[18] C. Samson, M. Le Borgne, and B. Espiau. *Robot Control: the Task Function Approach*. Clarendon Press, Oxford, United Kingdom, 1991.

[19] C. Thomas, S. Donikian. Virtual humans animation in informed urban environments. In *IEEE Computer Animation, CA'00*, pages 112–121, Los Alamitos, May 2000.

[20] X. Tu, D. Terzopoulos. Artificial fishes: Physics, locomotion, perception, behavior. In *proc of SIGGRAPH'94* , pages 43–50, Orlando, Florida, July 1994.

[21] B. Webber and N. Badler. Animation through reactions, transition nets and plans. In *International Workshop on Human Interface Technology*, Aizu, Japan, October 1995.

[22] C. Welman. Inverse kinematics and geometric constraints for articulated figure manipulation. Master's thesis, Simon Frasier University, September 1993.

[23] D. Zeltzer. Task-level graphical simulation : Abstraction, representation, and control, in *Making Them Move : mechanics, control and animation of articulated figures*, pages 3–33. 1990.