

Agent Architecture for a Real World Autonomous Virtual Guide: Interaction between the Decision and Perception Processes and Environment Representation



Morgan Veyret, Eric Maisel and Jacques Tisseau

European University of Brittany
National Engineering School of Brest
Computer Science Laboratory for Complex Systems
European Center for Virtual Reality, 25 rue Claude Chappe 29280 Plouzane, France1

Abstract—Museums like marine aquariums are facing a difficult problem when trying to deliver information to their visitors. The exhibits they propose are dynamic by definition. Each may contain multiple autonomous entities that need to be described to the visitor. Classical communication means (panels, audio-guides ...) are static and do not adapt to the constraints of such exhibits. We propose to use an autonomous virtual guide embedded inside the environment in order to describe it to the visitors. To describe this environment and the entities it contains, the agent must be able to perceive it. Doing so is a challenging task due to the dynamic and non-predictable aspects of this environment. In this article, we propose an architecture able to build a partial representation of such an environment that tends to be the “best possible representation” depending on the ongoing task. This is possible by setting up an interaction loop between the perception and decision processes of our intelligent agent. We describe this architecture and provide some results showing how this interaction effectively takes place in an experimental environment and in a real application setting.

Index Terms—Vision and scene understanding-architecture and control structure-distributed artificial intelligence-intelligent agents.

I INTRODUCTION

During the past ten years, museums and related structures (zoo, aquarium...) have evolved from simple artefacts, collections repositories to some sort of cultural attraction and visitor-centered exhibits. This evolution leads them to the installation and setup of new ways to communicate with the visitor [11]. The visitor’s behaviour also evolved and nowadays new cultural experiences are expected.

The parallel evolution of technologies with computers and multimedia in particular has provided interesting communication means between museums and their visitors. The World Wide Web and multimedia CD-ROMs [4] gave the opportunity to reach more people and to provide interactive multimedia presentations about the museum’s exhibits. Portable devices and ubiquitous computing introduced the

notion of adaptivity in the information presentation process, allowing that information to be tailored to the visitor’s location and/or preferences [24, 22]. Robotics also stepped into the cultural learning world through multiple tour-guide robots [6, 16]. Virtual and augmented reality have also been used as a new way to provide information to the visitor in the real [20] or virtual [28] museum.

Despite all these propositions, visitors usually prefer the historical mean of communication in museums: the human guide. This guide accompanies the visitor during their exploration of the exhibit, emphasising on specific ideas and facts. He also adapts his explanations to the audience and unexpected events.



Fig. 1. A visitor facing the aquarium. On the sides you can see the panels on which basic information about some of the species are provided. Linking the content of these panels to the actual content of the aquarium may be difficult and due to their limited size, only some of the fishes are presented. See Color Plate 1.

The central role of the human guide is even crucial in specific museum like the one we’re interested in (aquariums), where the exhibit is not just a collection of static objects but a real environment with autonomous entities that have to be described. Classical communication means like panels with text and pictures or audio-guides are generally used there. However, these standard supports are inappropriate in the particular case of the description of a real dynamic and unpredictable

environment. First, the information presented on these supports is static, i.e. they are not adapted to the actual content of the aquarium. This can be really problematic as, in the case of text panels for example, visitors have to constantly link the provided information to visible fishes. Moreover, the actual space available to place these text panels is limited and not all fishes can be described, and a visitor may be unable to find information about a particular fish he's interested in.



Fig.2. Visitor's view of the virtual guide. The virtual character is embodied into the real aquarium using augmented reality techniques. See Color Plate 2.

Here we present an autonomous virtual guide embodied into the environment (the aquarium) that is able to navigate and describe visible fishes to the visitors. This guide is integrated inside the real environment using augmented reality technology and provides information to the visitor using verbal and non-verbal modalities. To do so, the virtual guide must be able to perceive the environment in real time and to deliver coherent explanations that are both tailored to the dynamic environment (i.e. the virtual guide talks about the fishes visible to the visitor) and structured in a visit providing introductory information, details about a particular species, anecdotes and subject to some external constraints, like a maximum duration for example.

This article first presents some related work through a quick overview of the proposed architecture. Then we present some related work concerning visual perception. Section IV Part I briefly describes the animation system we use for our virtual guide. Section IV Part II then layout the decision process of the virtual guide, emphasising on how we select the right subjects depending on the environment and internal state of the guide while Section IV Part III presents the perception part of our architecture that aims at building the best possible representation in order to support the guide's explanations. Section IV Part IV then shows how these two parts of the architecture interact through the representation of the environment. Finally, Section VI describes experiments pointing out the two paths of the interaction loop.

II OVERVIEW

The idea of using an autonomous intelligent agent as a virtual guide is not new. As we said in the introduction of this document, museums are always looking for new ways to build exhibits in an entertaining and pedagogical way. This search

leads to some interesting ideas like web museums [19], virtual museums [7], information kiosks or mobile adaptive guides [2] providing new ideas concerning user adaptation, collaborative learning and education. In the field of autonomous agents, tour-guide robots [6, 15, 17] and conversational agents [10, 28, 9] have also been used widely as a tool for education and entertainment. Research dealing with the idea of a virtual guide usually focus on information delivery, user adaptation or content management.

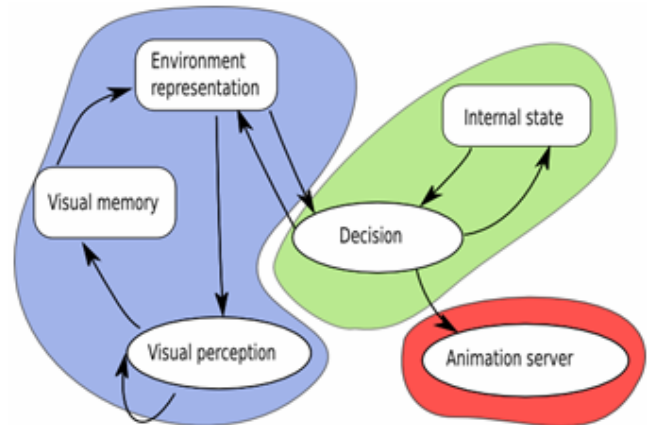


Fig. 3. Overall agent architecture. It follows the classic autonomous agent decomposition (perception, decision and action loop) but introduce some new interaction path between the perception and decision process through the environment representation.

What is noticeable in our work is the particular setting in which it takes place. The virtual guide is embodied inside a real dynamic and non-predictable environment which it must describe in real time in the form of a structured and coherent discourse. It should be able to decide what to tell to the visitor about this environment while being embedded in this environment in a credible way. The environment is perceived through a set of video cameras² placed in front of the environment.

In order to deliver relevant information to the visitor in real time the guide agent must be able to perceive the real world and select explanations according to this perception. To do so in a dynamic and unpredictable environment is tricky. The entities our guide must describe are autonomous and moving, constantly getting in and out of the field of view of the visitor which is a reference for the guide's explanations. Thus these explanations have to be interrupted gracefully keeping intact the overall discourse structure and coherence. Moreover, the representation of the environment can not be complete and accurate at every moment in real time due to the computational complexity of computer vision algorithms.

The overall architecture of our autonomous agent follows the classical perception-decision-action loop organization. However, to cope with our constraints, we need to be able to direct the perception process based on the decisions of our guide which are themselves guided by its perceptions thus setting up an interaction loop between the perception and decision processes.

² We can't put marker on the autonomous entities we'd like to perceive, especially if these entities a real living animals like the fishes of our application.

This loop is handled through the representation of the environment which operates as an interface between the perception and decision modules of our agent. The representation acts as a database containing information about the environment (entities along with their type, position and size) which the decision process can query to get needed information. Based on this information a subject is selected for explanation to the visitor through the scheduling of appropriate actions³.

This article focuses on this interaction loop. In the next section we provide an overview of the work related to this partial representation problem. Then we describe the two interacting parts of our system.

III RELATED WORK

Classic computer vision approaches tend to follow Marr's [18] view of visual perception. In this view, visual perception is simply the processing of visual information (images from the cameras in our case) in order to build a three dimensional representation of the environment on which cognition can act. Gibson [12] proposed an opposite view in his ecological view of perception. In this view, the world is here to be perceived. No information processing is required, instead he argues for the idea of direct perception.

In the field of artificial intelligence and autonomous agents, the problem of the necessity to have a representation of the agent environment or not has been debated a lot. Classical approaches inspired from the cognitive view of mind argue for a complete representation of the environment on which one can make inferences. On the opposite, direct perception followers argue that a representation is not required and probably useless.

What appears from these discussions is that a full reconstruction of the environment into a symbolic representation which is used by the decision process for inference purposes is not always practical (mainly due to computational complexity when facing real world cases) and probably not necessary as demonstrated by studies concerning direct perception [5].

However, our agent still needs a way to decide which explanation should be delivered to the visitor. And this can't be done directly from the images provided by the cameras. As Tsotsos demonstrated formally [25], building a complete and accurate reconstruction of the environment from images is too complex for a classic computer. So we need to find a compromise between the computational complexity of the visual perception problem and the need of a representation that may be used to take decisions.

A similar problem can be found in primate's visual system. Biological evidences have shown that our memory can handle only a limited amount of items at one specific time [13]. However we still have the feeling that we perceive our entire environment. This is possible because of attention as defined by [14]. Attention, and in our particular case visual attention, is what enables the selection of relevant information: since all

incoming information can't be processed in time, the solution is to select only the relevant ones. This idea is known as name of active perception [8] and has been in use in computer vision as well as in other computer science fields since many years [3, 26]. Along with this idea of selecting which information is relevant to the ongoing task and adapting the perception process to perceive this information, the notion of partial (or just-in-time) representation has also been inspired by biological evidences and the fact that our working memory is limited and usually contains only relevant information [27].

The architecture we present in this article is inspired by these ideas concerning visual attention and the notion of partial representation.

IV AUTONOMOUS AGENT ARCHITECTURE

As we described in the overview of our system, our architecture globally follows the classical perception-decision -approach of building agents. However, we propose a special interaction between the decision and perception process through the use of the representation of the environment. This representation is built by the perception and used by the decision.

In this section, we first describe briefly the action part of our autonomous agent. Then we go through the details of the decision and perception parts before dealing with how we realized the interaction between these parts.

4.1 Virtual character animation and action management

The animation of our virtual guide is viewed as a separate process controlled through the network using a simple command line style text protocol. This animation server handles the details of animations like navigation or lips synchronization when the guide talks. However it provides a comprehensive interface adapted to the information presentation goal of our application.

Navigation is handled using simple steering algorithms with the OpenSteer [1] library. For this purpose, the animation server receives the positions and sizes of all perceived fishes from the perception module. Different navigation modes are available:

- Monologue where the virtual character comes in front of the visitor. This mode is used to introduce the guide to the visitor and sometimes during the visit depending on the information to be presented.
- Waiting where the guide keeps its current position in the environment but still avoids obstacles.
- Follow where the guide avoid obstacles while moving to a specific position in the environment. When it reaches this position, the animated character moves around it. This mode is used to follow fishes during the explanation process.

The character is always running one of these three navigation modes. Meanwhile, it is able to respond to a set of specific commands that can be used for information presentation. These commands may be separated in two categories: animation and multimedia document presentation.

The first category allows us to trigger local animations like

³ Here actions refers to high-level atomic action like playing a sound file, triggering a specific animation or setting a navigation target to the animated character representing the guide as described in Section IV Part I.

an eye blink or fin movement. Those local animations may be started in parallel and are added to the current navigation animations. They are used to add some “personality” to the guide during information presentation.

The second category concerns multimedia documents playing. These documents may be pictures, videos or sound files and are used to actually provide information to the visitor. The implemented protocol allows us to start/stop the display of documents while enabling simultaneous playing of visual and audio documents.

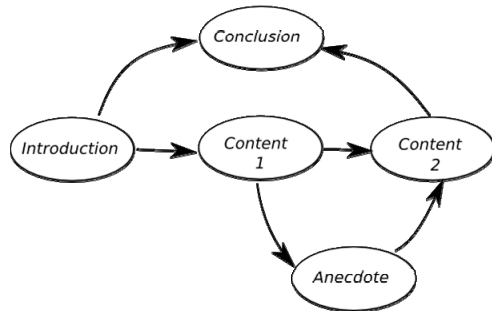


Fig. 4. Discourse scenario example. Each node is an atomic discourse element (i.e. a pre-recorded sound file) which can't be interrupted. Arrows show a possible path between two discourse elements. Multiple paths are allowed and selection is done at run-time based on the visit history, remaining time and/or visitor's interest.

We chose to use pre-recorded text parts as, to be entertaining, our guide may use some humoristic intonations and the like and we found that speech synthesis technology wasn't enough “real” to allow for such things. Based on these pre-recorded sound files, we defined a notion of discourse element which is the building block of our guide's discourse. Each discourse element corresponds to the playing of a specific sound file along with local animations and synchronization information. These elements are organised together in a graph-like structure we call a scenario (see Fig. 4). Each scenario contains information about a specific subject (e.g. shark reproduction) which is related to a number of topics (e.g. shark, reproduction, movements, food). Different paths are allowed inside a scenario. A specific path may be chosen at run-time based on the current visit status (discourse history, remaining time, visitor's interest ...). Moving from a scenario to another one is possible if a specific transition has been defined. A transition is a discourse element associated with a predicate function describing the possible use of the transition. This predicate may check the preceding scenario and/or the current visit status to decide if the transition is valid or not.

The guide's discourse is then a sequence of subjects that are selected by the decision process.

4.2 Decision

The decision process, which handles what subject is selected for explanation as well as how the virtual guide behaves in the environment.

Due to the specific context of our study (a virtual guide), we identified a small number of stages in the guide's life-cycle. Before describing these stages, we need to describe how the virtual guide's application will be used in a real setting. The

guide is running continuously, seamlessly integrated into the real environment through the use of a semi-transparent glass. When a visitor comes in front of this glass, the visit starts. Each visit is planned to be approximately five minutes long to increase the system availability for other visitors. A visit takes place as follows:

- First the guide introduces itself and the environment (the aquarium in our application).
- Then it starts the visit, choosing an appropriate discourse subject based on its internal state and the current environment representation.
- When the time is over, the guide concludes the visit and calls for the exploration of the rest of the museum.

These steps have been decided in cooperation with people who are responsible of defining the guide's behavior and discourse. Even if we try to build an autonomous virtual guide, these aspects (local behavior and discourse) have to be controlled by the authors of the visit. These authors may decide how the guide should present a specific topic or how it should behave in the environment. The decision architecture we propose here thus tries to give maximum control to the authors while it lets the virtual guide take the final decision of which behavior to adopt and which subject to talk about.

Looking at this typical use of our virtual guide, we are able to identify the following stages in its life-cycle:

- Idle stage, when there is no visitor; the guide should however exhibit some behavior increasing its credibility, making it more “life-like” as if it was really an entity inhabiting the real environment.
- Introduction stage, when the guide presents itself to the visitor.
- Visit stage, when the guide delivers explanations to the visitor. This is the main part of its life-cycle and we'll focus on this one in the remainder of this section.
- Conclusion stage, when the guide ends the visit in a proper way.

The virtual guide is always in one of these four stages which are organized accordingly to the state machine presented in Fig. 5.

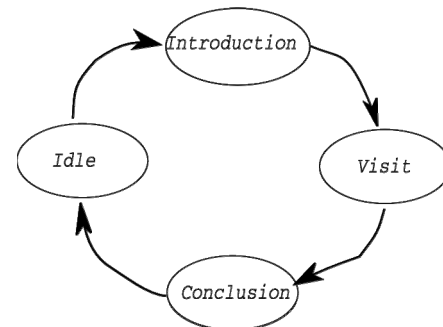


Fig. 5. The virtual guide's life-cycle. Each stage is used to decide the behavior of the guide in a set of existing interruptible state machines.

In each of these states, a behavior is selected and run. A behavior is a state machine with the particularity to be interruptible in specific conditions (see Fig. 6). The behavior management is not the focus of this article, it is however

important to note that a behavior can be interrupted by specific events only in a small subset of its states. During a visit, the entity being described can disappear or a more interesting entity may enter the visitor's field of view. The virtual guide should be able to cope with these changes gracefully without interrupting at random points in its current behavior and/or explanation. Defining possible interruption points in the behaviour allows the authors to know exactly when the guide may switch from a behavior to another one or when it may start talking about a new subject.

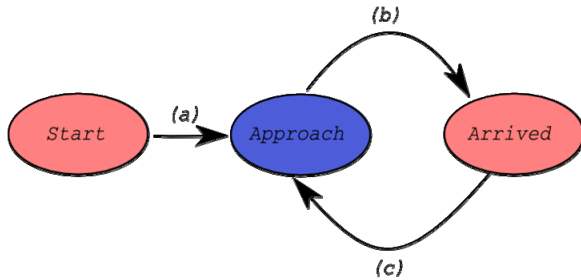


Fig. 6. A behavior. Blue state denotes a non-interruptible state. In this idling behavior, the virtual guide selects a position into the environment and move towards this position. While moving, the guide can not be interrupted, meaning its behavior can't be changed.

The main step of our virtual guide's life-cycle is the "explanation" one. During this step, the guide builds a discourse to deliver information to the visitor. To build this discourse, it selects the more interesting subject among all available. Then it selects an appropriate behavior to talk about this subject. Some possible behaviour may specify how the guide navigates into the environment and if it can be interrupted during its explanation about this subject. The behavior selection depends on the authors' desire (i.e. how they want the visit to look like) and the selected topic. For the moment, the behavior is simply selected randomly in a specific list depending on the current and previous discourse subjects.

The discourse subject is selected from a list of available topics. This list contains all topics as defined by the authors that have not yet been explained to the visitor. The selection process is as follow:

- First we select all subjects that may be used. This includes all defined subjects that have not yet been explained. Each subject is assigned an interest value of 0.
- Then this list of subject is fed into a succession of voting functions which modify the interest of each subject based on its viewpoint. Each voting function represents a specific point of view on the interest of a given subject regarding some given parameters. Some example functions are: a function giving fixed interest to a subject based on the authors assumptions (e.g. in the marine aquarium application, the shark subject is more interesting than other subjects); a function decreasing interest of a subject that has been explained recently; a function that increases the interest of the currently selected subject (i.e. to ensure the guide won't switch from a subject to another each time it's possible); or a function that increases the interest of a subject if a corresponding visual entity is present in the

environment representation.

- After the voting process the subject which got the highest interest value is selected. If several subjects have the same interest value, one is selected randomly. We can also specify additional constraints in this final selection process depending on the authors' needs. This can for example be a constraint ensuring that we won't select a subject that isn't associated with any visual entity. Although this can be done using the voting functions, those can not ensure that a subject won't be selected (think about the case where only one subject is available), and so we provide these post-processing constraints to allow the authors to specify "hard" constraints on the selection process.

Once a discourse subject has been selected, an explanation behaviour is selected. As we described earlier, for the moment this behavior selection only depends on the last subject and the current one, i.e. we select a different explanation behavior if we have to make a transition between a previously explained subject and the current one. This specific behavior may ensure that the guide doesn't select another subject immediately after the transition which would break the discourse coherence and structure.

When the explanation behavior is selected, it runs until the current subject is exhausted, or an event that may require a subject change occurs. These events are:

- A new visual entity appeared in the environment representation.
- The currently selected visual entity disappeared.
- The time allowed for the subject is elapsed.

These events restart the subject and explanation behavior selection process when the current guide state allows an interruption. This means that the guide has to be in an interruptible state to switch to a different subject.

Here we won't describe the discourse and decision process in more details since our focus is on the interaction loop between the perception and decision modules of our autonomous agent we're going to describe now.

4.3 Perception

The biggest part of our work deals with the guide's perception of the environment. In order to deliver appropriate information to the visitor, our guide must be able to perceive its environment or at least to be able to get some information, such as visible entities along with their positions in the field of view, visible size and type.

To do so, a representation of the environment is continuously built by the perception process of our agent. This perception process acts as an autonomous entity which extracts information from images from the visitor point of view of the real environment provided by the cameras.

The representation contains so-called *visual entities* each one being able to hold a set of properties including (but not limited to):

- Its size or more precisely the size of its bounding box.
- The position of this bounding box in the visible environment.

- Its type, which in our marine aquarium application correspond to the fish species.

The role of the perception process is to create and maintain these visible entities in the representation and to fill up their properties. This is achieved using a modular computer vision architecture composed of multiple “active objects” cooperating in the process of building the environment representation. Each object is assigned a specific task in this process, transmitting or requesting data to other objects using messages. Objects provide services for specific tasks that may be required by others. Objects may be distributed over multiple processes on the same and/or different computers seamlessly since they only rely on service discovery and messages functionalities. The perception module then consists of the set of all running objects at a specific time.

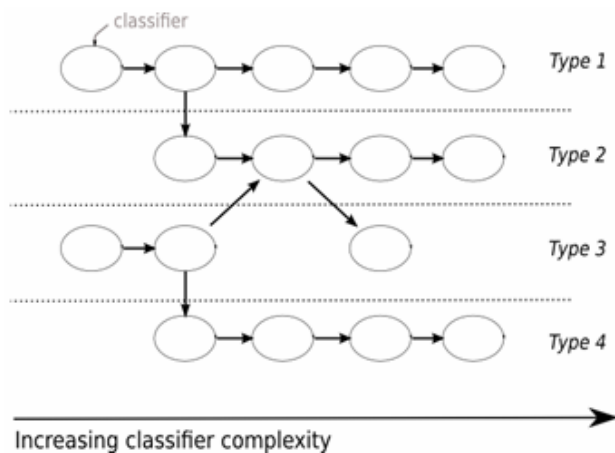


Fig. 7. Classifiers organization. Each ellipse is a specific classifier object. These classifiers are organized as sequences, each one matching a possible entity type. Classifiers may be shared between those sequences, resulting in a directed acyclic graph describing recognition paths.

All objects are in theory equal in the place they got within the perception module. Each object is able to communicate with other, there's no limitation in the requests or data that can be transmitted between objects, the only requirement being that an object requesting a specific service for another object must know how to request this service (how the request message looks like) and how to interpret the resulting data (what is contained inside the reply). However, from an architecture point of view, we are able to distinguish between the following objects categories:

- Representation/interface object. Only one instance of such an object is supposed to exist in the whole perception module. Its role is to interface between the perception module architecture (active objects and messages) and the decision module. Translating the simple communication protocol we use into a set of messages which must be understandable to the destination active objects. This object is also responsible for gathering all visual entities along with their properties in order to create the representation of the environment used by the decision process.
- Perception control objects. These objects are in charge of organizing the perception process by selecting and

initiating appropriate computations to create and maintain the environment representation. There are three of these, namely the vigilance, tracking and recognition objects. Each one is described in more details later.

- Utility objects. These objects handle some computations shared by multiple objects like the “entity decomposer” one which is able to break down a visual entity into several meaningful parts such as, in the marine aquarium application, the head, tail and body parts of a fish. Other noticeable utility objects are the image processing one which abstracts vision algorithms and low-level computations in the perception system and the classification ones which are used by the recognition object to compute the type of a specific visual entity.

All these objects cooperate to build the best possible representation of the environment. The main objects of the perception module are the “perception control objects”. There are three objects of this type:

- The *vigilance* object, which is responsible for detecting new visual entities, i.e. it creates new visual entities in the representation.
- The *tracking* object, which tries to maintain some visual entities properties related to its movement and geometry (e.g. position, size, trajectory ...).
- The *recognition* object, which manages the execution of available classification objects in order to set the “type” property of visual entities.

These objects rely on lower level services provided by intermediate objects as well as on the special image processing object.

These objects are pro-active, meaning that they may start self initiated computation, i.e. they do not only reply to request messages, but take the “initiative” of requesting computations to other objects.

The vigilance object requests specific image processing computations from time to time trying to detect salient areas in the field of view. To do so, it is provided a set of specific image-level properties to look for in specific image areas as well as time constraints on these computations (minimum and maximum delays). Every entity detected by the vigilance object is sent to the tracking object to reach a persistent state in the representation, e.g. to exist in the environment representation. This vigilance process makes possible to detect incoming entities in the visual field so the representation can contain recently appeared entities. Obviously, detecting new visual entities has a cost, it requires computations and can't be done continuously⁴. This is where the interaction between the perception and decision module first appears as we'll describe in Section IV Part IV.

The tracking object tries to maintain up to date as much visual entities as possible or more specifically it tries to maintain their dynamic properties such as their position, apparent size, speed or trajectory. It works using its internal memory containing the latest information concerning visual

⁴ Depending on the application and the particularities of the entities we consider, it may also be useless since new entities may only appear every 5 second or 10 minutes for example.

entities including both existing ones and those sent by the vigilance object. At each time step, the tracking object requests necessary computation to update stored entities. However, due to real-time constraints the overall number of computation is limited. Consequently the number of up to date entities the representation can hold at a given time is also limited and the perception module needs a way to select which entity to update first. This is again where the interaction loop will help us (see Section IV Part IV).

```
(get-visual-entities (entity-type "shark"))
```

Fig. 8. Example of an environment representation query. This will return all existing visual entities of type "shark" while increasing the perception interest for this type of entities.

The tracking and vigilance objects may be considered as building a general layout of the environment. The information they provide are volatile and continuously changing thus requiring frequent updates (especially for the tracking object). The recognition object works on a longer task: recognition of visual entities types. This is usually a more complex task requiring heavy computations. This is even truer when dealing with hard to recognize entities like the fishes of our application⁵. Moreover, our aim here adds another constraint to the "classic" ones of computer vision. We need to know the type of the visual entities to be able to describe them; it is not acceptable for the guide to deliver information about a specific entity while pointing at a misclassified one. Thus our objective here is not only to recognize these entities, but also to be sure of our recognition result so we don't deliver false information to the visitor. To cope with these hard constraints, the recognition object manages a set of classification objects (taking those that are available at run-time), each classifier returning a confidence level for a specific target to be of a specific type. The recognition objects manage multiple sequences of such objects organizing them as a directed acyclic graph (see Fig. 7). Each classifier may be a member of multiple sequences allowing results sharing and classifiers complexity (i.e. computation time) increase as you get further in the same sequence. Recognition is achieved by testing visual entities against a specific sequence, the type of an entity is known when a sequence has been successfully completed. If an entity fails in a particular sequence, it may be feed to another sequence or, if it has failed against all existing sequences, put back in the recognition queue to be processed again later. We won't get in more details concerning this process here since vision-based recognition is not the matter of this paper. However it is important to note how classifiers are used in this lines architecture. As we said, we needs to be sure about an entity type before talking about it. Following this requirement, we view the recognition of an entity as ensuring that this entity E is of type T instead of trying to find the type of this entity among all possible types. Then we need to choose the type to test the entity against. This is again where the interaction loops is

useful as we'll see in the next Section IV Part IV.

Since this is not a computer vision article, we won't detail the image processing module or the internals of the classifier and we'll continue by describing how interaction between the decision and perception processes is achieved.

4.4 Perception-Decision interaction

Due to the computational complexity of the computer vision task in a open and dynamic environment such as the one we're interested in, the task of creating and maintaining an environment representation in real time is a challenging one.

Detecting, tracking and recognizing visual entities from simple images is not easy and requires a lot of computations. Computational resources are limited and even with new powerful computers, the computations needed to reconstruct the entire environment are not achievable in real time⁶. It is also not possible to do these computations when the decision process needs them since those requires some time to be completed.

```
(define-visual-property-test
 entity-type (a-type)
 '(equal (visual-entity-type
         current-request-entity)
        ,a-type))
(define-visual-property-control
 entity-type (a-type)
 '(increase-type-interest ,a-type))
```

Fig. 9. The two functions associated with the query in Fig. 8. The first one returns the entities matching the provided type in the current environment representation while the second one increase interests associated with this particular entity type.

Consequently, the perception process we described in Section IV Part III does not try to build a full and accurate representation of the real environment instead, nor does it try to gather information when requested by the decision process. Instead, it focuses on providing the "best possible representation" of the environment. To do so, the perception module must know about what will be needed by the decision process as well as when it will be needed.

We make this possible using the perception architecture presented in Section IV Part III: an entity that autonomously builds a partial representation of the environment. However, to be useful to the decision process and to become what we call the "best possible representation", this entity needs to know about the ongoing decision process to adapt its computations to the current needs of our agent.

This is where our proposed architecture makes sense and differs from the classical perception/decision/action loop that is common to most autonomous agents' architectures. What we propose is a continuous interaction between the decision and perception processes, meaning that the decision process is influenced by the contents of the environment representation while this content depends on the current needs of the agent.

⁵ The task of recognizing fishes to be of a specific species is hard firstly because of the lighting conditions of an aquarium (caustics). Second fishes are deformable objects which can be viewed by the camera at different angles depending on their position and/or orientation/movement.

⁶ This is also not necessary (not desirable?) because only a small subset of the information available in this environment is relevant to the explanation task and decision process of the agent.

To achieve such a loop, we make direct use of the representation of the environment. This representation is accessible to the decision process as a database. In fact all accesses to the content of the representation have to be done using database-style queries as show in Fig. 8.

Such a query will return matching entities in the current state of the environment representation and it will also increase related interests in the perception module. These interests are how the perception module is controlled and how it knows what is needed by the decision module. We have identified different types of interest in the perception process:

- Visual entities interest.
- Entities type interest.
- Areas interest.

Interests are used by the perception control objects to decide which computation to do first consequently increasing the chances to get a result for this computation since all computations can't be done at a given time due to the real time constraint.

Interests associated with a specific query are defined as prior knowledge. For example, a query as the one show in Fig. 8 will increase interest for the entity type "shark" as well and the entities interest of all visual entities currently matching that type. It can also increase the interest for properties and areas that are known to be associated with sharks in the image like uniform color areas and middle level water. The definition of this knowledge is done when specifying possible representation queries. The `get-visual-entities` function alone is unable to query the environment representation. Instead it just translates queries into a succession of computations to do and messages to pass to the perception module based on the fields present in the query. When defining a field we wish to be able to query (like entity-type for example), we in fact define the two aspects of an environment query:

- The test returning all matching entities in the current representation.
- The control procedure which takes care of increasing the necessary interests in the perception module

As an example, the query presented in Fig. 8 will translate into two main calls: one matching entities in the current environment representation, the other one taking care of sending the appropriate interest control messages. These calls are defined as show in Fig. 9.

Then when the decision process makes a query in the environment representation searching directly for specific information, this query returns the current matching entity in the representation and asks the perception module to increase its interest in specific information related to this query, therefore directing the perception process in computing needed information. The representation queries are done in the decision process each time it needs information about the environment especially during subject selection when looking for example for entities related to a specific subject.

V RESULTS

We have conducted some experiments on our control

architecture. We studied the two aspects of the interaction loop: the influence of the decision process on the perception and the influence of the perception process on the decision.

The first aspect has been studied using artificial data. We ran our perception system on a video containing colored squares as you can see in Fig. 10. These squares represent the autonomous entities we'd like to describe. Each square is moving autonomously and has a continuous movement (e.g. they don't jump from one location to another). There are three types of "artificial" entities: "red", "green" and "blue" named after their color.

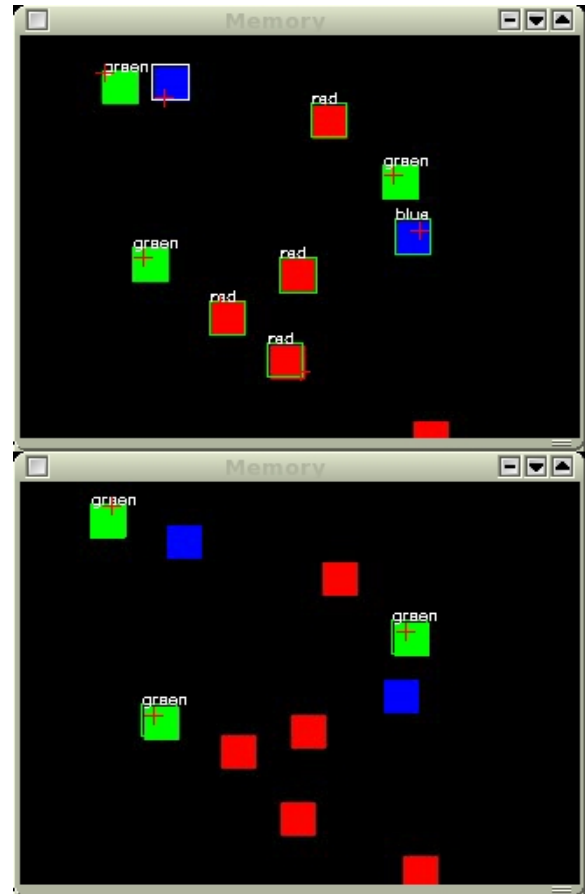


Fig. 10. A screen-shot of the memory content overlapped on the video we used. The test video contains multiple colored squares that move continuously in all directions. On the top picture you can see the content of the representation at a specific time without any external influence on the perception process. Entities of different types are detected and recognized. On the bottom one, you can see the environment representation content at the same time but with a high interest on "green" targets.

During the test, we measured the total number of visual entities in the environment representation as well as the number of entities for each type. We also measured the value of the interest in the recognition, tracking and vigilance perception control objects.

Fig. 11 shows a plot of the number of entities as well as the interest in the recognition object during the test. We only plotted the interest in one module here to show how it is link with the content of the environment representation. In other objects (namely tracking and vigilance objects), interest values are varying in a similar way. Since this test shows the influence

when requesting a specific entity type the recognition interest is the more significant one. The test was executed as follow:

- During the first part (from 0 to 300) the perception module was running without any external influence.
- From 300 to 450, we applied an external influence by requesting entities of type “green” to the environment representation.
- At 450, we stopped the external influence on the perception module.

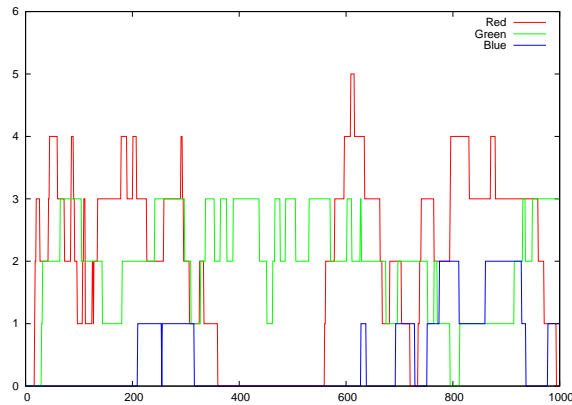


Fig. 11. Influence of external perception control on the content of the memory. On this graph you can see the total number of detected entities as well as the number of entities for each type. The left y axis unit is the number of entities. The x axis correspond to perception system updates, it depends on the computer running the test. In this case one time step is approximately 83 milliseconds (12 frames per second).

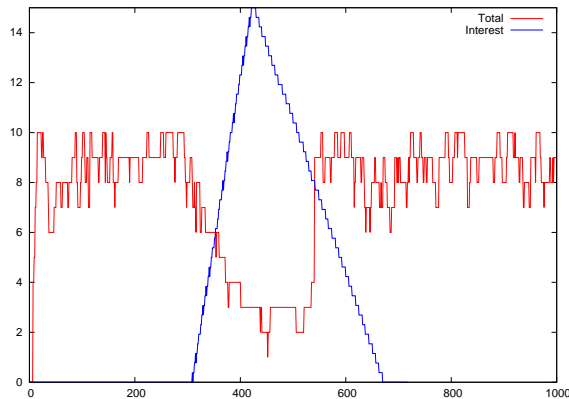


Fig. 12. The graph with a high peak describes the evolution of the interest for targets of type green. Its highest value is around 40 but has been scaled to fit on the same graph as the memory content. The other graph represents the total number of tracked visual entities.

During the first part of the test you can see in Fig. 11 and Fig. 12 that the perception module normal functioning mode is to try to build a “complete” representation of the environment. When there is no interest, each entity, type and areas are treated in a similar way (i.e. everything has a chance to be computed). On a computer with enough computational power, this functioning mode would lead to a complete representation of the environment.

When the external influence is started, the interest for entities of type “green” is increased. The external influence we applied was intentionally high. This has been done to show clearly how

the interest changes the content of the environment representation. As the interest increases (the high peak on Fig. 12) all perception objects start to focus on this particular entity type. The vigilance object decrease the detection time for the color green while the tracking object computes properties of the “green” entities first and the recognition object tries the “green” type classifier before all other possibilities. As a result, the total number of entities contained in the environment representation decreases while the number of “green” entities remains constant and matches the total number of entities of this type available.

When we stop the external control of perception (around 450) the value of interest starts to decrease as the whole perception system tends to get to a stable state (it tries to equalize interest values). Meanwhile, the total number of detected entities as well as the number of “red” and “blue” entities increases to finally get back to the initial state of the system.

This test is just here to show how the perception module acts as an autonomous entity trying to build a representation of the environment without any prior knowledge. It treats all processes in a similar way since it can’t decide if information in the representation is useful or not. Then requests on the representation content implicitly give information to the perception module, telling it what is relevant to the decision process at a specific moment. This perception module then adapts its computations and tries to compute information suited to this requirement. As the Fig. 10 shows, without any external influence the perception module is able to detect entities of different types while under the influence of a request concerning entities of type “green”, the environment representation contains only entities of this particular type. This behavior comes from the really high external influence we applied on the perception system using the request shown in Fig. 12. This is not the “normal” behavior but we think that such an extreme case is better to show clearly how external influence modifies the content of the representation.

The second aspect we studied is the influence of the perception process on the decision. To study this aspect we use a real case coming from our application to the french marine museum Océanopolis [21]. Since this aspect is hard to describe in a textual form, we ran a simple test to show how subject selection may be influenced by the content of the environment representation. Other aspects of the decision process may be influenced by the content of the environment representation like which behavior to select for example.

In this test, the virtual guide is able to talk about the following subject: the aquarium, the guide (itself), platax (a specific species) and sharks. This test was done using an environment representation previously stored in a file. In this representation, there are first no identified entities then an entity of type “platax” is detected.

All subjects may be used without any particular entity in the representation. Subject selection is done using the following voting functions:

- One which sets a static priority of +10 to subject aquarium and guide and +5 to subject platax and shark.
- One which adds a +20 priority to a subject when an associated entity has been detected.

During the test we measured the priority of each subject. When there were no specific entities in the environment representation, subjects' priorities where:

- aquarium: +10
- guide: +10
- platax: +5
- shark: +5

The aquarium subject was selected and run. Consequently the subject was removed from the list of available ones. The remaining subjects got similar priorities as described previously until the entity of type "platax" was detected. This detection allowed the second voting function to increase the interest of the subject platax associated with this particular entity type. Leading to the following list of priorities:

- platax: +25
- guide: +10
- shark: +5

Then the subject platax was selected. The simple example provided here highlights only a small part of the influence of the environment representation on the decision process. But we consider this influence as obvious since, using computer vision to perceive the real environment; the virtual guide perception is done solely through the environment representation built by the perception module. Thus only information contained in this representation is available to the guide when it searches for information about the environment.

VI CONCLUSION & PERSPECTIVES

We have presented our architecture for an autonomous virtual guide embedded in a real environment containing real life autonomous entities (fishes in our case). We described our architecture for such an autonomous agent focusing on an interaction loop between the perception and decision process. This interaction enables the construction of a partial environment representation which is influenced by the ongoing decision process that uses this representation. We presented some results showing how this interaction works through two experiments focusing on the two possible paths of the interaction.

This work is obviously not complete and many research paths are open. Some of them may be the modification of the perception module to build it as a real multi-agent system based on the extension of the interest idea. For the moment we're focusing on building a real world application as well as providing accurate results about the system.

ACKNOWLEDGEMENT

The authors wish to thank Océanopolis for their cooperation in building a real world application for this research.

REFERENCES

- [1] Opensteer-steering Behaviors for Autonomous Characters. <http://opensteer.sourceforge.net>.

- [2] G. D. Abowd, C. G. Atkeson, J. Hong, S. Long, R. Kooper and M. Pinkerton. Cyberguide: A Mobile Context-aware Tour Guide, *Wireless Networks*, vol. 3, pp. 421-433, 1997.
- [3] R. Bajcsy. Active Perception, *Proceedings of the IEEE, Special issue on Computer Vision*, vol. 76, no. 8, August 1988.
- [4] D. Bearman. Hands-on: A 1995 Snapshot of the Evolution of Interactive Multimedia, in *Third International Conference on Hypermedia and Interactivity in Museums (ICHIM 95 / MCN 95)*, 1995.
- [5] R. Brooks. Architectures for Intelligence, Chapter How to Build Complete Creatures Rather than Isolated Cognitive Simulators, pp. 225-239. Lawrence Erlbaum Associates, Hillsdale, NJ, 1991.
- [6] W. Burgard, A. Cremers, D. Fox, D. F'ahnel, G. Lakemeyer, D. Schulz, W. Steiner and S. Thrun. The Interactive Museum Tourguide Robot, in *proceeding of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)*, 1998.
- [7] L. Chittaro, L. Ieronutti and R. Ranon. Navigating 3d Virtual Environments by Following Embodied Agents: A Proposal and Its Informal Evaluation on a Virtual Museum Application, *PsychNology*, vol. 2, pp. 24-42, 2004.
- [8] P. Churchland, V. S. Ramachandran and T. Sejnowski. Large-scale Neuronal Theories of the Brain, *Chapter A Critique of Pure Vision*, MIT Press, 1994.
- [9] P. de Almeida and S. Yokoi. Interactive Character as a Virtual Tour Guide to an Online Museum Exhibit, *Museums and the web 2003*.
- [10] P. Doyle and K. Isbister. Touring Machines: Guide Agents for Sharing Stories about Digital Places, 1999.
- [11] M. A. Drake. *Encyclopedia of Library and Information Science, Chapter Museum Informatics*, pp. 1906-1913, CRC Press, 2003.
- [12] J. Gibson. The Ecological Approach to Visual Perception, Houghton Mifflin, 1979.
- [13] M. M. Hayhoe, D. H. Ballard, H. S. Jochen J. Triesch, P. Aivar and B. T. Sullivan. Vision in Natural and Virtual Environments, *Eye Tracking Research and Applications Symposium*, 2002.
- [14] W. James. The principles of psychology, 1890.
- [15] G. Kim, W. Chung, K.-R. Kim, M. Kim, S. Han and R. H. Shim. The Autonomous Tour-guide Robot Jinny, in *proceedings of 2004 IEEWRSJ International Conference on Intelligent Robots and Systems*, 2004.
- [16] K. Kim, T. H. Chalidabhongse, D. Harwood and L. Davis. Background Modeling and Subtraction by Codebook Construction, in *ICIP 2004*.



Morgan Veyret is currently doing a PhD in Computer Science at the European Center for Virtual Reality (CERV). His research interests are in virtual/augmented reality and the perception architectures of autonomous characters in these environments.



Eric Maisel received his PhD in Computer Science in 1992 from the University of Rennes I. His research interests are in virtual reality, augmented reality, perception of virtual autonomous characters and their application in museography. He is currently lecturer at ENIB, an engineering school in France.



Jacques Tisseau was born in 1953. He is a professor in Computer Science and is head of LISyC and CERV. His domains of interest are entity's autonomy in virtual environments, and "in virtuo" experimentation in the modeling processes.