



HAL
open science

Visualisation adaptative d'un système complexe : le cas du corps humain

Yohann Gélisse

► **To cite this version:**

Yohann Gélisse. Visualisation adaptative d'un système complexe : le cas du corps humain. Synthèse d'image et réalité virtuelle [cs.GR]. 2011. dumas-00636398

HAL Id: dumas-00636398

<https://dumas.ccsd.cnrs.fr/dumas-00636398v1>

Submitted on 27 Oct 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Rapport de stage
master informatique spécialité Recherche en informatique

Visualisation adaptative d'un système complexe le cas du corps humain

Yohann GELISSE
École Nationale d'Ingénieurs de Brest

Encadrant : Eric MAISEL
Equipe ARéVi
Centre Européen de Réalité Virtuelle (CERV)
Technopôle Brest-Iroise
29280 Plouzané

Juin 2011

Résumé

Ce rapport détaille différents moyens de mise en œuvre d'un outil à effet pédagogique. Le but du rapport étant d'identifier les méthodes les mieux adaptées pour la construction d'un atlas du corps humain interactif. Le rapport présente tout d'abord un état de l'art assez large dans différents domaines à prendre en compte pour le développement d'un tel système. Un deuxième chapitre présente les principes retenus pour mettre en œuvre différentes solutions. Un troisième chapitre détaille certains points importants dans les choix effectués pour le développement des principes précédents. Finalement, le rapport permet de conclure sur les différentes solutions les plus appropriées à la construction d'un système interactif d'exploration du corps humain.

Table des matières

1	Introduction	3
1.1	Contexte	3
1.2	Objectifs	4
2	État de l’art	5
2.1	Développement d’un outil d’enseignement	5
2.2	Description du corps humain	6
2.2.1	Unified Medical Language System (UMLS)	6
2.2.2	Foundational Model of Anatomy (FMA)	6
2.3	Filtrage des objets	8
2.3.1	Méthodes géométriques	8
2.3.2	Modèle d’interaction spatiale	8
2.3.3	Système hybride	9
2.4	Annotation de modèles 3D	12
2.4.1	Annotations 3D	12
2.4.2	Modèle de connaissances	13
2.4.3	Placement des annotations	14
2.5	Génération d’explications	16
2.5.1	Graphes d’explication	16
2.5.2	Mémoire du guide	17
2.5.3	Structure à base de schéma	18
2.6	Identification des intentions du visiteur	21
2.6.1	Musée portatif	21
2.6.2	Types de visiteur	21
2.6.3	Réseau Bayésien	21
3	Principes	24
3.1	Vue générale	24
3.2	Module de visualisation	24
3.2.1	Présentation	24
3.2.2	Graphe de scène	25
3.2.3	Instructions de visite	26
3.2.4	Réaction à des événements	26
3.3	Différents guides	27
3.3.1	Guide interactif	27
3.3.2	Guide directif	27
3.3.3	Guide pro-actif	30

4	Mise en œuvre	32
4.1	Visualisation	32
4.1.1	Commandes de visite	32
4.1.2	Environnement	32
4.1.3	Capteurs	34
4.2	Planification d'explications	36
4.2.1	Graphe de représentation du corps humain	36
4.2.2	Réseau Bayésien	36
4.3	Interaction visualisation/planification	36
4.3.1	Communication par socket	36
4.3.2	Langage de communication	36
5	Conclusion	38
6	Perspectives	40
6.1	Utilisation de modèles 3D	40
6.1.1	Kinect	40
6.1.2	Modélisation du Visible Human Project	41
6.2	Affichage des annotations	44
6.3	Amélioration des capteurs de proximité	44
6.4	Expérimentation	44

Chapitre 1

Introduction

1.1 Contexte

Le corps humain est complexe. On comprend alors la difficulté qui se pose aux personnes chargées de l'enseigner, plus particulièrement dans le domaine des professions de santé. Il faut présenter une vision claire et structurée à un public hétérogène. Pour cela, les atlas anatomiques au format papier et les fichiers multimédia sont d'une aide précieuse mais présentent malgré tout des inconvénients : il n'est en effet pas toujours facile de les comprendre (en particulier à cause du point de vue forcé qu'ils imposent, comme illustré par la figure 1.1), il y a peu, voire pas du tout d'interaction avec l'utilisateur et l'immersion de ce dernier est faible. La réalité virtuelle semble alors être une base intéressante pour développer des applications telles qu'un atlas anatomique numérique interactif et immersif, pour accroître l'efficacité de l'apprentissage. Mais un système tel que le corps humain, composé d'un grand nombre d'éléments aux formes diverses, fait naître le problème de la complexité des données à traiter.

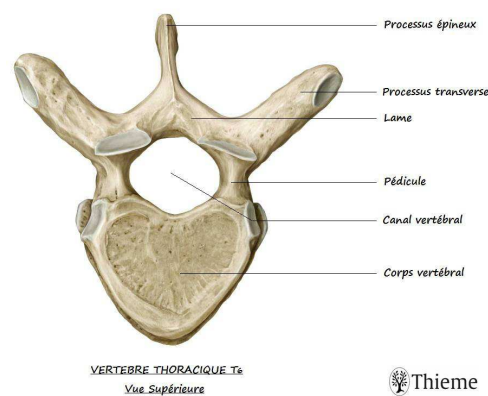


FIGURE 1.1 – Exemple d'image visible dans un atlas anatomique papier

1.2 Objectifs

Afin de répondre aux problèmes présentés dans la section précédente, nous allons nous intéresser à la mise en œuvre d'un outil d'aide à la compréhension d'objets et de phénomènes complexes. Ce type de système se doit de n'afficher à l'utilisateur que les objets nécessaires et les informations correspondantes, tout cela au bon moment, afin de rendre les présentations plus claires.

Pour cela, il est nécessaire de mettre en place un filtrage des objets à afficher qui respecte une certaine logique liée aux déplacements de l'utilisateur et ses objectifs.

Une fois le tri des objets réalisé, des informations les concernant doivent être ajoutées pour permettre à l'utilisateur de mieux se situer dans les éléments qu'il connaît, mais également de découvrir de nouvelles choses. Ces informations seront disponibles sous la forme d'annotations qui peuvent être de différents types (texte, audio, vidéo,...) afin d'avoir une large gamme de possibilités d'apprentissage. On peut définir un modèle d'annotation 3D qui permet de prendre en compte les subtilités de l'espace 3D mais également le classement des informations selon des concepts liés au domaine d'étude.

En complément de ces annotations, un guide virtuel (visible ou non) peut permettre de présenter les différentes informations à l'utilisateur en suivant un texte prédéfini ou en rassemblant différents bouts d'explications dans une logique compréhensible par l'utilisateur.

Pour mener à bien la mise en place de cet environnement virtuel pédagogique, il sera donc nécessaire d'implémenter ces différentes techniques de manière à assurer leur complémentarité.

Chapitre 2

État de l'art

2.1 Développement d'un outil d'enseignement

Chittaro propose un article assez général [1] sur l'utilisation des technologies Web3D pour le développement d'outils pédagogiques. Dans notre cas, nous n'allons pas utiliser ce genre de technologies mais il reste néanmoins intéressant à lire pour bien cerner les environnements virtuels éducatifs (EVE). Certains avantages et besoins y sont soulignés comme :

- L'interaction.
- La nécessité d'une navigation simple.
- Une expérience multisensorielle.
- Plus de points de vue qu'avec une représentation 2D.

Ce sont différents points que nous allons essayer de mettre en œuvre ou tout du moins rendre possible dans le développement de notre outil pédagogique.

Un second article, écrit par Harry Brenton et al. [2] s'intéresse plus particulièrement à l'utilisation du multimédia et du Web3D pour l'enseignement de l'anatomie. Ce dernier commence par un rappel des différentes méthodes d'enseignement utilisées autre que l'utilisation de modèles 3D. Plus loin, il détaille les différentes techniques qui permettent de modéliser un objet en 3D, ce qui pourra nous être utile en perspective de ce stage (voir section 6.1). Avant de présenter les deux ressources Web3D que les auteurs sont en train de développer, ils posent les avantages et inconvénients des modèles 3D.

Avantages

- Donnent différents points de vue à l'utilisateur.
- Permettent la représentation du temps.

Inconvénients

- Leur mise en œuvre chronophage.
- Le besoin d'ordinateurs assez puissants pour les présenter.
- L'interaction avec les modèles fortement liée aux périphériques utilisés.

Le reste de l'article s'intéresse aux environnements qu'ils étaient en train de mettre en œuvre, lors de l'écriture, pour l'apprentissage de la structure et du fonctionnement du plexus brachial (nerfs du membre supérieur).

2.2 Description du corps humain

2.2.1 Unified Medical Language System (UMLS)

L'UMLS [3] est l'unification de plusieurs vocabulaires spécialisés dans les sciences médicales. Il a été créé afin de faciliter l'accès aux différentes ressources existantes dans le domaine, et également pour homogénéiser les différents moyens d'exprimer un même concept. Son utilisation est gratuite dans le cadre de travaux académiques. Nous nous y sommes donc inscrit pour pouvoir y naviguer.

L'UMLS est divisé en trois parties : le Métathésaurus, le réseau sémantique et le lexique SPECIALIST.

Le Métathésaurus est la base de l'UMLS. Il contient plus d'un million de concepts médicaux originaires d'une centaine de vocabulaires différents. Parmi eux, on retrouve le Foundational Model of Anatomy (FMA) dont nous allons parler dans la section suivante.

Chaque concept du Métathésaurus est lié à une ou plusieurs catégories du réseau sémantique. Ce dernier contient 135 catégories différentes (ex : objet, structure, événement) et 54 types de relations entre concepts (ex : X est un Y, X est physiquement relié à Y, X peut causer Y).

Enfin, le lexique SPECIALIST relie différentes variantes d'un même concept en une seule expression afin de permettre d'homogénéiser l'UMLS (ex : infarctus du myocarde est une variante de crise cardiaque).

2.2.2 Foundational Model of Anatomy (FMA)

Le FMA [4] est un des vocabulaires utilisés par l'UMLS. Il s'agit d'une ontologie spécialisée dans l'anatomie humaine. Elle est complète et peut donc être utilisée et simplifiée pour le développement d'applications. Elle propose une hiérarchie des différents éléments du corps humain (voir figure 2.1). Chaque élément dispose d'une description composée de relations avec d'autres éléments comme :

- X est membre de Y
- X a pour extrémité Y
- X est dans la continuité de Y

- Human body
 - ✦ Head
 - ✦ Body proper
 - ✦ Upper limb
 - ✦ Right upper limb
 - ✦ Left upper limb
 - ✦ Lower limb
 - ✦ Right lower limb
 - ✦ Left lower limb
 - ✦ Integument
 - ✦ Muscular system
 - ✦ Skeleton
 - ✦ Set of all joints
 - ✦ Set of all viscera
 - ✦ Neuraxis
 - ✦ Vasculature of body
 - ✦ Set of all nerves
 - ✦ Thoracic duct tree
 - ✦ Right lymphatic duct tree
 - ✦ Dental arcade
 - ✦ Musculoskeletal system
 - ✦ Nervous system
- Respiratory system
 - ✦ Respiratory tract
 - ✦ Right pleural sac
 - ✦ Left pleural sac
 - Set of intercostal muscles
 - ✦ Diaphragm
 - Neural network of respiratory system
 - Vasculature of respiratory system
 - ▣ Lower respiratory tract
 - ✦ Tracheobronchial tree
 - ✦ Left lung
 - ✦ Right lung
 - ✦ Lung

FIGURE 2.1 – Exploration de la hiérarchie du FMA

2.3 Filtrage des objets

Un environnement virtuel n'a réellement de l'intérêt que s'il prend en compte les objectifs de l'utilisateur en n'affichant que les objets qui sont pertinents. Effectivement, si l'environnement ne s'adapte pas à la demande de l'utilisateur, ce dernier risque de s'y perdre ou de ne pas atteindre ses objectifs. Pour mettre en place un système de filtrage efficace, il est nécessaire de définir les variables qui permettent d'estimer la pertinence ou non d'un objet selon la position de l'utilisateur et les objectifs qu'il doit suivre. Dans cette démarche, différentes approches ont été proposées.

2.3.1 Méthodes géométriques

Une première méthode basique pour gérer le filtrage des objets consiste à se focaliser sur des propriétés géométriques telles que la distance ou la visibilité d'un objet par rapport à la position de l'utilisateur dans l'environnement virtuel. C'est le cas de nombreux travaux mettant en jeu des acteurs virtuels (Reynolds [5],...) qui perçoivent leur environnement au moyen de capteurs virtuels basés sur des algorithmes de découpage d'espace ou de lancer de rayon.

L'avantage de cette méthode est qu'elle repose sur des critères simples à évaluer. La complexité de calcul est d'autant plus faible que ces méthodes utilisent des structures de données pour optimiser leurs calculs. Cependant, trop d'objets non pertinents risquent d'être sélectionnés ou à l'inverse, d'autres objets pertinents ne seront pas gardés (voir figure 2.2a). En effet, comme l'indique Julier [6], l'importance d'un objet n'est pas toujours assimilable à des propriétés telles que la distance qui le sépare de l'utilisateur.

2.3.2 Modèle d'interaction spatiale

Benford et Fahlén [7] proposent un moyen de se baser sur les méthodes géométriques en les définissant pour chaque objet, à travers leur Modèle d'interaction spatiale (Spatial Model of Interaction). Ce modèle, initialement conçu pour un environnement virtuel distribué, permet de déterminer les utilisateurs (ou plus généralement les objets) qui interagissent entre eux et permet également de définir le niveau d'attention que chacun a pour un autre. Pour cela, les auteurs définissent la notion de médium, qui représente les moyens de communication habituels (audio, texte,...) ou tout autre moyen spécifique d'interfaçage entre objets. Ensuite, pour chaque médium utilisé, différents sous-espaces existent :

Aura : sous-espace qui délimite la zone dans laquelle l'objet concerné peut potentiellement interagir avec un autre. Si l'aura de deux objets entre en collision, ils peuvent interagir.

Focus : sous-espace d'attention. Plus l'objet B est dans le focus de l'objet A, plus A est attentif à B.

« [...] *the more an object is within your focus, the more aware you are of it* [...] »

Nimbus : sous-espace de présence. Plus l'objet A est dans le nimbus de l'objet B, plus A est attentif à B.

« [...] *the more an object is within your nimbus, the more aware it is of you.* »

Le niveau d'attention d'un objet pour un autre est donc fonction de leur focus et de leur nimbus.

La notion d'adaptateur permet d'utiliser des objets pour amplifier ou diminuer l'aura, le focus ou le nimbus d'un objet. Par exemple, dans un médium audio, le microphone permettrait à un objet d'augmenter son aura et son nimbus.



(a) Basé sur la distance



(b) Avec le modèle d'interaction spatiale

FIGURE 2.2 – Résultats avec deux systèmes de filtrage différents

Julier [6] propose d'adapter ce concept pour faire en sorte qu'un utilisateur puisse être attentif à un objet qui se situe loin de lui, en augmentant le nimbus de cet objet. Pour illustrer cette approche, il utilise un système complexe constitué par une ville de 30 bâtiments et plus de 740 objets distincts. Le but de cette application est que l'utilisateur puisse s'y déplacer en étant guidé par la mise en surbrillance de l'objectif à atteindre (ici, une pièce dans un immeuble). En utilisant un critère de distance (comme proposé dans la section 2.3.1), un grand nombre d'objets est visualisé (voir figure 2.2a). En se basant sur l'approche de Julier, c'est à dire en donnant un nimbus important aux objets concernés par l'objectif, seul le bâtiment que doit atteindre l'utilisateur et sa porte d'entrée sont mis en surbrillance (voir figure 2.2b). On a ici une première forme de filtrage intéressante.

2.3.3 Système hybride

Un système plus général de filtrage d'informations consiste à utiliser une base de règles, à partir de données telles que la position et l'orientation de l'utilisateur, les relations d'occlusion entre les objets ou encore l'importance d'un objet dans la tâche effectuée par l'utilisateur. Le

problème de cette approche est qu'elle montre vite ses limites lorsqu'il s'agit de gérer de nombreux objets. En effet, s'il faut gérer les relations entre chaque objet (ou entre l'avatar de l'utilisateur et chaque objet), le nombre de relations augmente rapidement.

Pour réduire la complexité de calcul, Julier [6] présente un système hybride de filtrage de l'information, qui consiste à faire un premier tri des objets à l'aide du modèle d'interaction spatiale pour ensuite travailler sur l'ensemble réduit d'objets. Cette approche repose sur les trois étapes suivantes :

1. Initialisation : Calcul du focus de l'utilisateur et du nimbus pour chaque objet en fonction des objectifs de l'utilisateur. Cette initialisation doit être effectuée lorsque les propriétés des objets ou les objectifs de l'utilisateur changent.
2. Sélection des objets dont le nimbus entre en collision avec le focus de l'utilisateur (simplification du modèle d'interaction spatiale).
3. Affinage éventuel de la sélection en utilisant une logique de plus haut niveau (base de règles par exemple).

Pour pouvoir effectuer la première étape, il faut être capable de représenter les objectifs de l'utilisateur, l'importance de chaque objet pour cet objectif précis, et donc en déduire un processus de calcul du focus et des nimbus. Afin d'y arriver, deux nouvelles notions ont été définies :

Les propriétés objectives sont les propriétés d'un objet identiques pour chaque utilisateur (par exemple la taille ou la position d'un objet).

Les propriétés subjectives sont les propriétés d'un objet qui dépendent de l'utilisateur et de ses objectifs. Elles sont représentées sous la forme d'un vecteur d'importance.

De plus, un vecteur de tâche est affecté à chaque utilisateur pour permettre d'avoir plusieurs objectifs à la fois. Ce vecteur contient chacune des tâches et une valeur de pertinence par rapport aux activités courantes de l'utilisateur. Ainsi, il est possible d'assigner des priorités aux tâches en leur attribuant des poids différents.

Exemple

Pour illustrer cette approche, nous allons présenter l'exemple utilisé par Julier et al. dans un article plus approfondi que le premier [8]. Cet exemple concerne l'évitement de snipers par un soldat. Comme dans la plupart des cas, les tâches pertinentes de l'utilisateur (le soldat) ont été définies par des experts du domaine (ici, l'armée américaine) et sont les suivantes :

1. Parcours : L'utilisateur passe d'un point à un autre de l'environnement.
2. Entrée dans un immeuble : L'utilisateur entre dans un immeuble qu'il suspecte habité par un sniper.
3. Planification stratégique : L'utilisateur planifie un itinéraire large entre différents bâtiments.
4. Planification tactique : L'utilisateur planifie un itinéraire dans une zone réduite (peu de bâtiments).
5. Offensive : L'utilisateur effectue des manœuvres offensives dans le but de désarmer un sniper.
6. Défensive : L'utilisateur effectue des manœuvres défensives en réponse à l'attaque d'un sniper.

Ces différentes tâches forment le vecteur de tâche de l'utilisateur.

Comme expliqué auparavant, chaque objet possède également un vecteur d'importance qui a été défini comme suit :

1. Position connue d'un sniper
2. Endroit qui peut être utilisé par un sniper pour préparer une embuscade
3. Protection contre les attaques des snipers
4. Endroit qui peut être utilisé pour rassembler des civils
5. Endroit fréquenté par des alliés
6. Repère suivi par un utilisateur lors d'un itinéraire
7. Mis en avant manuellement par l'utilisateur

Le focus de l'utilisateur est défini manuellement comme un carré autour de sa position dont les dimensions peuvent varier de 0 à 500m.

Le nimbus de chaque objet est défini en fonction de son état, de l'état de l'utilisateur et du médium concerné. Ainsi, la dimension du côté d du nimbus d'un objet i pour un utilisateur j , à travers le médium m est le suivant :

$$n_{i,d}^{j,m} = (v_d^j + z_d^j) * (1.0 + \zeta_i^{j,m})$$

Avec v_d^j et z_d^j les tailles de l'objet et de sa zone d'impact dans la dimension d et $\zeta_i^{j,m}$ une valeur d'inflation liée à l'objet et à la tâche de l'utilisateur. Sa valeur est calculée à l'aide de la formule suivante :

$$\zeta_i^{j,m} = 0.2 \min(1, (M_j)^T * s_i^j)$$

où s_i^j est le vecteur des propriétés subjectives de l'objet et M une matrice construite en projetant le vecteur de tâche dans le vecteur d'importance, ce qui donne pour notre exemple :

$$M = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

qui signifie, pour la 4^e ligne par exemple, que le système montrera la position des civils à l'utilisateur (4^e élément du vecteur d'importance) seulement dans le cas où sa tâche est la planification stratégique ou tactique (3^e et 4^e éléments du vecteur de tâche).

Pour finir, l'interaction entre focus et nimbus est calculée de la manière suivante :

- interaction nulle si le focus et le nimbus ne se chevauchent pas.
- interaction totale si l'utilisateur est dans le nimbus de l'objet.
- interaction intermédiaire lorsque le focus et le nimbus se chevauchent (mais que l'utilisateur n'est pas dans le nimbus). Sa valeur est donc de :

$$(1.0 - \text{distance minimum entre utilisateur et bordure du nimbus}) / \text{longueur du focus}$$

Ce système a été évalué par différents utilisateurs à l'aide d'un système tel que celui présenté figure 2.3 et le résultat obtenu a été concluant. En effet, ces utilisateurs en ont ressorti que le système de filtrage supprimait bien les informations superflues tout en maintenant les informations essentielles à l'objectif d'évitement des snipers.



FIGURE 2.3 – Prototype mobile d'un système de réalité augmentée

2.4 Annotation de modèles 3D

2.4.1 Annotations 3D

Dans un environnement virtuel tridimensionnel, il peut être intéressant d'ajouter des annotations 3D pour apporter des connaissances supplémentaires à l'utilisateur ou pour le guider. Une annotation 3D a les mêmes propriétés qu'une annotation 2D habituelle, auxquelles s'ajoutent des éléments propres à la 3D. Lenne et al. [9] ont proposé un modèle d'annotation 3D sur lequel nous pourrions nous baser. Celui-ci se décompose en trois parties différentes qui sont présentées ci-dessous.

Forme

- La forme de l'annotation correspond à ses données concrètes. On y retrouve :
- le support utilisé pour diffuser l'information.
 - le contenu, qui est l'information diffusée par l'annotation.
 - la présentation, qui est la manière dont le contenu est présenté avec le support choisi.

Spatialisation

La spatialisation contient les données qui permettent de placer l'annotation dans l'espace 3D. On y trouve le point d'ancrage, qui permet de fixer l'annotation dans l'espace, soit en donnant les coordonnées géométriques (ancrage spatiale), soit en faisant référence au produit annoté (ancrage logique). On y trouve également un second aspect, propre à la 3D, le point de vue géométrique. En effet, dans un espace tridimensionnel, un objet peut facilement être vu sous plusieurs angles, ce qui fait qu'un aspect présenté par une annotation peut être visible sous un angle mais pas sous un autre. Cela peut donc induire en erreur l'utilisateur dans sa compréhension de l'information délivrée par l'annotation.

Méta-données

Les méta-données apportent des informations sur l'annotation. Elles peuvent être séparées en deux types :

les méta-données épisodiques qui concernent le contexte de création, de modification et de lecture de l'annotation, comme par exemple le nom d'une personne ou une date.

les méta-données sémantiques qui concernent plutôt le sens à donner à l'annotation (degré de priorité et de confiance, fonction argumentative de l'annotation, public visé et méta-données du contenu).

2.4.2 Modèle de connaissances

Lorsque le nombre d'annotations est trop élevé, il devient difficile pour l'utilisateur de lire les annotations, ou même au créateur de l'environnement de se rappeler si un aspect a déjà été abordé. C'est pour ces raisons que Lenne et al. [9] proposent d'utiliser un modèle de connaissances basé sur une ontologie pour indexer les annotations.

Ontologie

Aubry explique dans sa thèse [10] (qui approfondie l'article de Lenne) qu'une ontologie est un moyen de représenter des connaissances sous la forme de concepts liés par des relations, dans un domaine donné. Ces relations peuvent être d'ordre sémantique ou hiérarchique.

Dans le cas d'utilisation de Lenne, les annotations concernent un vélo observé par différents utilisateurs dans un environnement collaboratif. L'ontologie est donc composée de trois parties qui sont les suivantes :

- La partie principale concerne les concepts spécifiques au domaine des annotations, qui est donc ici la conception mécanique.
- Une seconde partie concerne la structure physique du produit. On y retrouve principalement des relations `part_of` entre les concepts.
- La troisième partie concerne les méthodes de conception.

Utilisation du modèle

Dans l'environnement présenté par Lenne, des concepts de l'ontologie sont proposés à l'utilisateur lorsque celui-ci crée une annotation, afin de permettre son indexation. Le choix des concepts est déterminé automatiquement par une simple analyse syntaxique du texte de l'annotation. L'utilisateur peut ainsi choisir un de ces concepts, ou d'autres en naviguant dans l'ontologie. Par la suite, le modèle de connaissances a seulement besoin d'utiliser les méta-données de contenu des annotations pour afficher les annotations qui répondent à un concept choisi.

Évaluation

Ce modèle de connaissances a été évalué auprès de 15 étudiants. Le but était de répondre à un questionnaire en utilisant l'environnement virtuel et ses annotations. Pour cela, les étudiants étaient séparés en deux groupes : K+ pour ceux qui avaient accès au modèle de connaissances et K- pour ceux qui n'y avaient pas accès. Les tâches à accomplir par les étudiants étaient du type "Montrer l'annotation mettant en avant le fait que l'attelage doit être perçu comme fiable". Les résultats obtenus par cette évaluation se sont montrés concluants. Le nombre de bonnes réponses est beaucoup plus élevé pour le groupe K+ (69% contre 27% pour le groupe K-). De

plus, le nombre de questions auxquelles les étudiants K+ accordent un haut niveau de confiance est également plus élevé que pour les K- (62% des réponses pour le groupe K+ contre 35% pour le K-). Un point négatif peut malgré tout être relevé. En effet, les utilisateurs K+ ont mis légèrement plus de temps à répondre au questionnaire que les K- (46 minutes contre 43). Ceci s'explique par le fait que la recherche des concepts dans le modèle de connaissances peut être plutôt longue. Mais comme nous l'avons vu, ce temps est largement rentabilisé.

2.4.3 Placement des annotations

Après avoir fait le choix des annotations à afficher, il faut s'occuper de la façon de les placer. Il est important que les annotations ne se cachent pas entre elles et qu'elles ne cachent pas un autre objet que celui qu'elles concernent. Bell et Feiner [11] proposent une approche en deux temps : il s'agit tout d'abord (1) de gérer le placement des annotations pour éviter qu'elles ne se chevauchent et ensuite (2) de faire en sorte que ce positionnement n'engendre pas de discontinuités dans l'affichage lorsque l'utilisateur se déplace.

Règle de positionnement

Cette approche permet de positionner les annotations de deux façons différentes : soit à l'intérieur de l'objet concerné, soit à l'extérieur (voir figure 2.4). La première possibilité concerne plutôt les annotations textuelles courtes (ou labels). En effet, les annotations multimédia (image, vidéo,...) vont souvent cacher partiellement ou entièrement l'objet qu'elles annotent. Elles devront donc être placées à l'extérieur de la projection de l'objet annoté.

Le calcul de placement se fait en considérant les rectangles qui englobent les projections des annotations, caractérisées par des informations 3D (cf. 2.4.1). Dans le cas d'un label (ou étiquette), le rectangle qui lui correspond dépend de la police voulue et de la taille choisie. L'utilisateur peut déterminer une gamme de tailles pour les étiquettes. Ainsi, lors du déroulement de l'algorithme de placement d'une étiquette, on vérifie dans un premier temps si l'objet concerné accepte les annotations internes. Si c'est le cas, on calcule le plus grand rectangle interne qui peut contenir la plus grande copie du label (en essayant les différentes tailles de la gamme).

Si jamais l'objet est plus grand, l'algorithme va essayer de déterminer une surface disponible pour insérer l'étiquette à l'extérieur de l'objet. C'est également cette étape qui est utilisée pour les annotations autres que les labels (car elles ne peuvent pas être insérées en interne). Dans cette étape, on détermine les plus larges rectangles d'espace libre dans la projection de l'environnement. Ensuite, on choisit, dans cet ensemble, un rectangle qui peut contenir l'annotation en restant proche de l'objet concerné, pour permettre de faire le lien entre les deux. Pour éviter toute ambiguïté, une flèche est ajoutée entre l'objet et l'annotation.

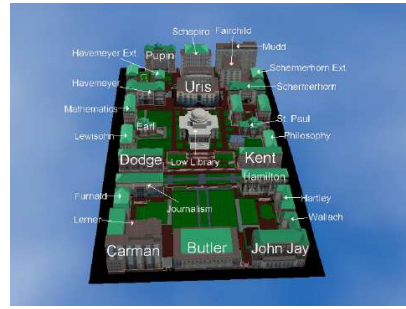
Continuité temporelle

Jusque-là, nous nous sommes intéressés au positionnement d'annotations sans parler de mouvements continus de l'utilisateur dans l'environnement. Ainsi, dans le cas actuel, si le positionnement est calculé à chaque déplacement de l'utilisateur, les annotations peuvent bouger de manière significative (oscillation entre deux états, évolution discontinue,...) et cela devient rapidement fatigant pour le visiteur. Pour gérer ce problème, Bell et Feiner introduisent trois autres concepts dans leur approche.

Tout d'abord, un état d'hystérésis permet d'empêcher une oscillation entre l'affichage et la disparition d'une annotation lorsque celle-ci est à une taille proche de sa taille minimale. Pour cela, trois contraintes sont définies sur la taille d'une annotation : sa taille maximale, sa taille



(a) Annotations externes seules



(b) Annotations internes et externes

FIGURE 2.4 – Annotation de modèles 3D

minimale et sa taille minimale absolue. Les changements dans l'annotation se font ensuite en respectant les règles suivantes (voir figure 2.5) :

- si l'espace disponible est supérieur à la taille minimale d'une annotation (\min), alors celle-ci est affichée ;
- si l'espace disponible est inférieur à la taille minimale absolue (absMin), alors l'annotation disparaît ;
- si une annotation est affichée avec une taille entre sa valeur minimale absolue et sa valeur minimale, alors elle est enlevée au bout d'un temps n ;
- dans les autres cas, l'objet est affiché à la taille la plus grande possible, sans dépasser sa taille maximale.

On peut remarquer que des règles similaires sont appliquées pour faire le choix entre un affichage à l'intérieur ou à l'extérieur d'un objet.

Pour cette méthode d'état d'hystérésis, c'est le choix du temps n et de la différence entre la taille minimale et la taille minimale absolue qui permettent d'éviter les oscillations.

Pour éviter les changements trop radicaux dans la position et la taille des annotations, une deuxième notion est mise en œuvre qui consiste à faire le choix entre le meilleur agencement possible de l'annotation à l'instant t et l'agencement le plus proche du précédent. Pour cela, les deux possibilités sont comparées à chaque fois. Si elles sont identiques, on l'applique. Sinon, on attend pendant une certaine durée et si les deux possibilités ne coïncident toujours pas lorsque ce temps est écoulé, on applique le meilleur agencement.

La dernière technique consiste à passer en douceur d'un agencement à l'autre lorsque nous appliquons les méthodes précédentes. Ainsi, une annotation sera interpolée de sa position précédente à la nouvelle sélectionnée. Également, lors du passage d'une annotation interne à une annotation externe, la taille de la flèche liée à l'objet grandit ou rétrécit.

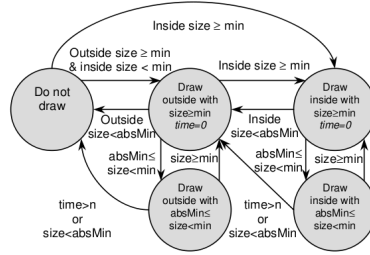


FIGURE 2.5 – État d’hystérésis pour l’affichage des annotations

2.5 Génération d’explications

Dans les deux chapitres précédents, nous nous sommes intéressés à des mécanismes capables de supprimer les objets inutiles à la compréhension d’un objet complexe et d’augmenter les objets restants en leur associant des annotations.

Ce chapitre introduit la notion du temps. Il ne s’agit plus de considérer l’objet à expliquer à un instant unique mais dans un intervalle de temps plus ou moins long pendant lequel cet objet est expliqué. La présentation d’un objet n’est alors plus considérée comme se déroulant sur une suite d’instantants indépendants mais comme un processus structuré.

Dans cette démarche, plusieurs approches sont possibles. Veyret [12] les a classées de la manière suivante :

visite statique : La structure de la visite est prédéterminée et ne peut être modifiée au cours de l’exécution.

visite interactive : La visite se structure à partir des actions du visiteur. C’est lui qui indique explicitement au système les objets ou les sujets pour lesquels il souhaite des explications.

visite réactive : Des explications sont automatiquement données au visiteur selon sa position et son orientation. L’utilisateur peut malgré tout demander explicitement plus de détails sur un objet précis.

visite dynamique : Le guide structure la visite au cours de son déroulement en fonction de l’environnement qui l’entoure. Il choisit les objets qu’il présente en fonction de leur distance ou de leur rôle dans l’environnement. La distinction avec les approches interactive et réactive est l’adaptation de la visite à l’initiative du guide et non plus en réaction aux actions du visiteur. Certaines approches [13] permettent également à l’utilisateur de se déplacer de manière autonome, auquel cas le guide l’accompagne.

Nous allons particulièrement nous intéresser à l’approche de visite dynamique, qui paraît plus appropriée pour une visite pédagogique qui laisse un peu de liberté à l’utilisateur.

Nous allons présenter différentes approches qui permettent de structurer l’information et qui utilisent un mécanisme permettant de lier ces informations pour former une explication cohérente.

2.5.1 Graphes d’explication

Le problème auquel s’intéressent Veyret et al. [14] est de donner à des visiteurs des explications relatives à un aquarium réel. Cet environnement est par nature dynamique et peu prévisible.

La solution retenue pour représenter le guide consiste à superposer l’image du guide virtuel (un modèle 3D animé de poisson) et celle de l’aquarium. Le guide virtuel donne alors l’impression

de se diriger vers un poisson intéressant et de le suivre tout en l'expliquant à l'aide de fichiers multimédia.

Outre les problèmes de perception de l'aquarium par le système, se pose celui des explications à donner. Que faire si le poisson (ou sujet) expliqué disparaît ? Que faire si un poisson plus intéressant apparaît dans le champ de vision du visiteur ?

Structure de l'information

Une explication pouvant être interrompue, il n'est pas possible de la représenter de façon monolithique. La solution retenue dans l'article consiste à formaliser l'explication par un graphe dont les sommets sont des éléments d'explication (fichiers multimédia, animations,...) et dont les arcs représentent la relation de précédence entre deux éléments. Chaque espèce de poisson correspond à un graphe.

Ainsi, produire une explication revient à trouver un chemin dans le graphe et à restituer les différents éléments d'explication qui le composent.

Construction de l'explication

Le parcours d'un graphe est contrôlé par un automate hiérarchique modélisant le comportement du guide virtuel. L'automate passe d'état en état selon la représentation de son environnement. À chaque transition correspond une action : se diriger vers un point ou un objet particulier, sélectionner le sommet suivant dans le graphe courant,... Il permet donc de contrôler à la fois l'évolution du guide mais également la production des explications.

De cette façon, l'automate permet de structurer les explications dans le temps : le guide commence par donner des explications très générales puis va donner des explications plus précises, en les alternant avec des anecdotes...

D'autre part, certains états sont interruptibles : Quand on les atteint, il est alors possible de réévaluer l'environnement afin de modifier le sujet à expliquer.

Cette approche permet ainsi d'obtenir un contrôle dans le temps de la production des explications dans un environnement dynamique.

2.5.2 Mémoire du guide

Ibanez [15] cherche à simuler un guide virtuel pour expliquer des sites relevant du patrimoine historique et culturel. Son approche consiste dans un premier temps à rassembler des éléments d'histoire pour former une histoire complète et dans un second temps, à transformer l'histoire de façon à laisser transparaître le point de vue du guide virtuel.

Structure de l'information

Chaque élément d'histoire est caractérisé par différents paramètres comme :

- un nom ;
- un type d'évènement, qui permet de classer les éléments ;
- une localisation, qui peut-être une position de l'environnement ou un objet spécifique ;

La connaissance est structurée sous la forme d'une mémoire du guide (voir figure 2.6a), composée :

- d'une liste de paires localisation/élément d'histoire ;
- d'une fonction d'évaluation des couples précédents ;

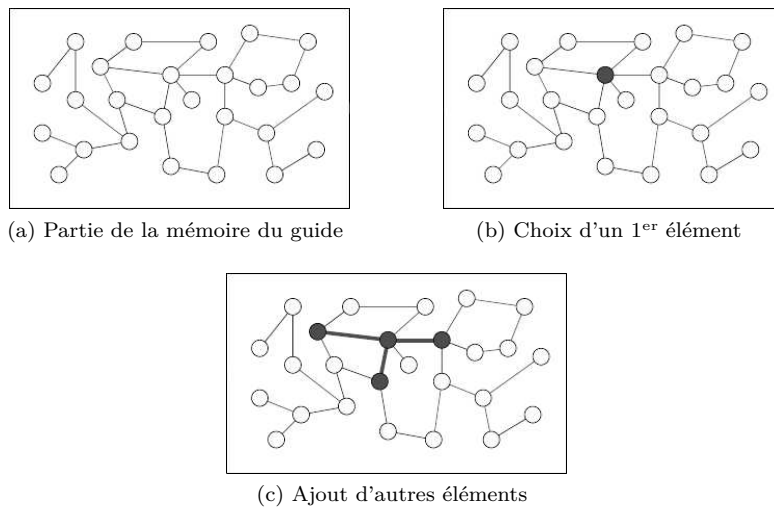


FIGURE 2.6 – Fonctionnement de la mémoire du guide

- de relations entre éléments d’histoire du type cause-effet et sujet-objet.

Construction de l’explication

À chaque étape de la visite, la meilleure paire lieu/élément est choisie (voir figure 2.6b) à l’aide de la fonction d’évaluation des paires. Cette dernière tient compte de la distance entre le lieu concerné et le dernier lieu visité, de la proximité des thèmes de l’histoire avec ceux précédemment abordés, et également de la préférence du guide pour un thème (en fonction de ses centres d’intérêts, son sexe, ses origines, sa religion,...). L’explication est ensuite étendue en sélectionnant d’autres paires liées à l’élément d’histoire choisi (voir figure 2.6c), jusqu’à obtenir une granularité satisfaisante.

Les éléments d’histoire choisis sont ensuite traduits par rapport à la perspective du guide (en contrôlant par exemple l’attitude du guide par rapport à ce qu’il présente). Des règles dites de bon sens permettent d’ajouter de nouvelles informations dans l’explication en se basant une nouvelle fois sur le profil du guide.

L’arbre ainsi obtenu est linéarisé et chacun des éléments d’histoire est associé à un texte pour transformer le tout en une histoire.

2.5.3 Structure à base de schéma

Yuan [13] s’est concentré sur la création d’un guide qui assiste l’utilisateur dans sa navigation à l’intérieur d’une galerie d’art virtuelle, tout en engageant des conversations sur les expositions et en réagissant de manière non-verbale en respectant les codes sociaux au niveau des gestes et des expressions faciales.

Structure de l'information

Dans cette approche, l'information est structurée sous forme de schéma. Un schéma est un modèle de discours représenté comme un réseau de déclarations courtes qui correspondent au type de narration voulu (par exemple "décrire") et au sujet qui doit être développé. Dans la figure 2.7a qui représente l'espace de discours, nous pouvons voir les déclarations courtes (points noirs) réunies sous forme de réseau pour former ainsi un schéma (ovale gris foncé). Plusieurs schémas sont ensuite réunis pour former une entité de domaine (ovale gris clair). Dans le cas de la galerie d'art virtuelle présentée par Yuan, une entité de domaine est par exemple un artiste.

Pour gérer l'aspect temporel de l'histoire à générer, les déclarations courtes peuvent être liées par une relation de séquençement pour gérer l'ordre des éléments au sein d'un même schéma, ou par des relations de changement de sujet et de référencement pour connecter des éléments de deux schémas différents.

Pour chaque déclaration courte, il est possible de définir des mots-clés ou des modèles de questions que pourrait poser l'utilisateur à ce sujet. Ceci permet de comprendre les demandes de l'utilisateur et ainsi de diriger le guide vers les explications qu'il doit donner en réponse.

Pour décrire une visite, deux types de plans sont définis (voir figure 2.7b) :

le plan de visite (tour plan) qui passe en revue les activités principales de la visite sous la forme d'un squelette. On y retrouve les remarques d'introduction de la visite et la séquence d'objets à présenter.

le plan de discours (discourse plan) qui décrit le contenu (c.-à-d. les schémas) de chaque activité.

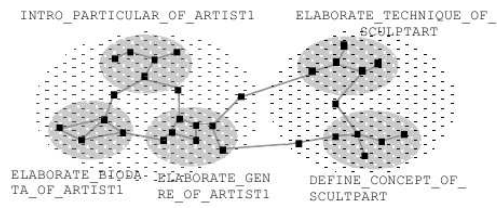
Construction de l'explication

La bibliothèque de plans contient tous les plans de visite et de discours. Il est précisé pour chacun d'eux le but qu'il permet d'atteindre.

À l'exécution, un module spécifique (le planificateur) choisit un plan en fonction de l'objectif du guide. Par exemple, si le visiteur se dirige vers un objet, il sera indiqué que l'objectif du guide est de présenter cet objet. Un plan de discours formulé dans ce but, et composé de différents schémas, pourra ainsi être choisi pour remplir cette tâche. Ainsi, chacun des sous-buts du plan sera exécuté pour remplir l'objectif du guide.

C'est le module appelé ordonnanceur qui gère cette action. Pour cela, ce dernier se place à un point d'entrée du schéma initial et se déplace au fur et à mesure jusqu'à traverser tous les schémas du plan, pour ainsi former une séquence des déclarations courtes à réciter.

Un autre module, de coordination du dialogue, permet de faire basculer le guide dans différents modes, soit pour naviguer à travers les schémas (initialisation, maintien et changement de schéma), soit pour interagir avec l'utilisateur. Si par exemple ce dernier manipule un objet ou s'éloigne, le guide passera dans un mode de blocage pendant quelques secondes. Si l'utilisateur pose beaucoup de questions au guide, celui-ci va interrompre ses explications et passer dans un mode de conversation pour attendre d'autres questions.



(a) Espace de discours

Tour Plan	Discourse Plan
welcome	INTRO PARTICULAR OF MYSELF
remarks	INTRO THEME OF EXHIBITION
appetizer	INTRO BACKGROUND OF GALLERY
	INTRO PARTICULAR OF ARTIST
	ELAB MILESTONES OF ARTIST
artifact#1	DESCRIBE FORM OF AF1
	DESCRIBE MATERIAL OF AF1
	Reference: DEFINE MATERIAL MATERI
	IDENTITY ARTISTINTENTION OF AF1
artifact#3	DESCRIBE FORM OF AF3
	DESCRIBE TEXTURE OF AF3
	DESCRIBE FEATUREPART OF AF3
	Probe
	IDENTITY ARTISTINTENTION OF AF3
	...

(b) Plan de visite et plan de discours

FIGURE 2.7 – Modèle de discours

2.6 Identification des intentions du visiteur

Cette partie se concentre sur deux articles trouvés dans le domaine, publiés par un même auteur. Dans celui-ci, Flavia Sparacino [16] [17] présente un système mis en œuvre dans un musée réel qui permet de cerner le profil du visiteur. L'objet de ces travaux est d'adapter la présentation du guide aux intentions du visiteur.

2.6.1 Musée portatif

Il s'agit d'un système de réalité augmentée que doit porter le visiteur (voir figure 2.8). Il permet de lui présenter des explications (texte, vidéo, image) en plus de ce qui est visible dans le musée. À cela s'ajoute un capteur infrarouge qui permet de détecter où se situe le visiteur et de chronométrer le temps pendant lequel il s'arrête à chaque endroit du musée. Tout un réseau d'émetteurs infrarouge est installé dans le musée pour permettre cela.

2.6.2 Types de visiteur

F. Sparacino s'est intéressée à différents travaux de recherche sur la classification des visiteurs de musée.

Eleanor Hooper-Greenhil [18] a identifié plusieurs groupes de visiteurs de musée : les familles, les groupes scolaires, les autres groupes éducatifs, les "apprenants par loisir", les touristes, les personnes âgées et les personnes à facultés physiques, sensorielles ou mentales réduites.

Dean [19], quant à lui, propose trois catégories plus larges de visiteurs :

- le visiteur occasionnel, qui traverse rapidement le musée s'en vraiment s'y impliquer.
- le visiteur sommaire, qui va s'arrêter plus particulièrement sur quelques objets pour y passer du temps et lire des informations.
- une troisième catégorie, une minorité, qui va passer énormément de temps dans le musée pour tout lire et examiner tous les objets.

Serrell [20] divise également les visiteurs en trois types : le visiteur de passage, "l'échantillonneur" (sampler viewer) et le méthodique.

F. Sparacino s'est surtout inspirée des travaux de Dean et Serrell pour finalement retenir trois principaux types de visiteur adaptés à ses travaux :

- busy ("occupé"), qui souhaite une vue d'ensemble des éléments principaux du musée.
- selective ("sélectif"), qui étudie en profondeur seulement quelques éléments du musée.
- greedy ("avide"), qui souhaite voir le plus de choses possible.

Dans le système retenu, l'estimation du type du visiteur est obtenue à l'aide d'un réseau Bayésien qui prend en entrée les données recueillies à l'aide des capteurs infrarouge présentés dans la section précédente.

2.6.3 Réseau Bayésien

Le réseau Bayésien (voir figure 2.9) permet principalement de faire ressortir le profil d'utilisateur. Ce réseau est composé de deux nœuds pour chaque objet : un premier pour la localisation de l'objet et un second pour la durée pendant lequel l'utilisateur s'y intéresse.

Chaque nœud de localisation est lié avec son voisin pour représenter une topologie du musée. Ce nœud permet également d'évaluer l'intérêt d'un objet. Celle-ci est exprimée en trois valeurs : neutre, intéressant, ennuyeux.

Pour le nœud de durée, il aurait été plus précis d'utiliser des valeurs continues, mais cela aurait engendré beaucoup plus de travail au niveau du réseau. Le choix s'est donc plutôt porté sur trois valeurs discrètes : temps négligeable, court, long. Pour pouvoir passer les valeurs réelles

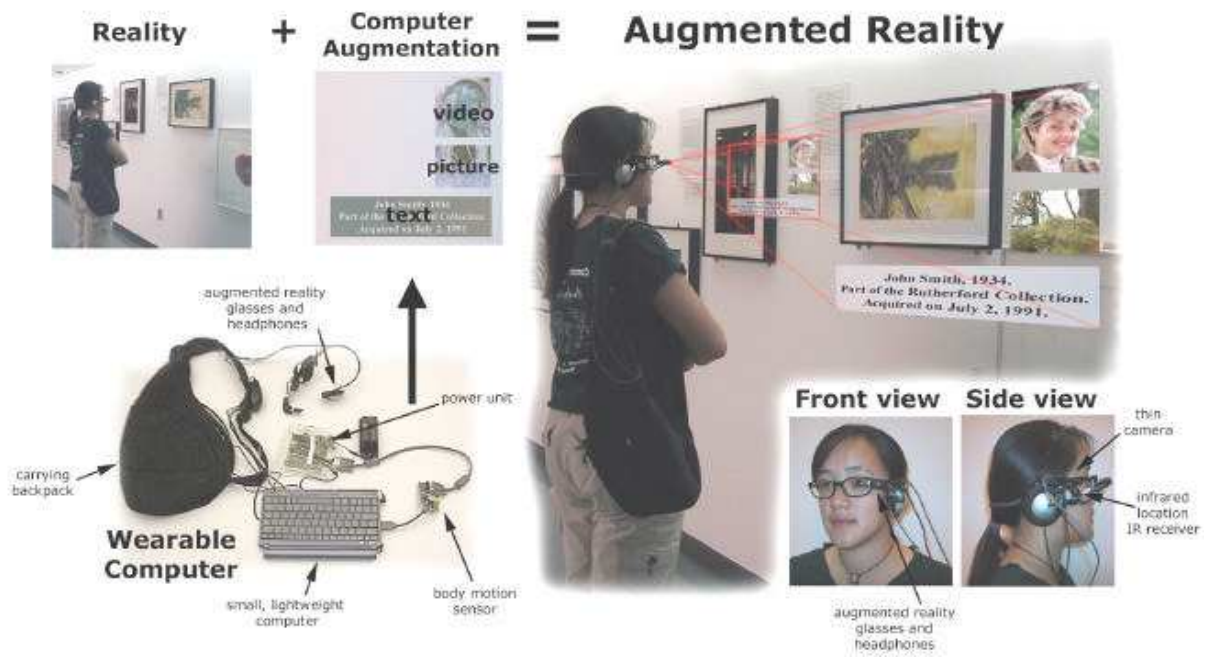


FIGURE 2.8 – Présentation du système portatif de réalité augmentée pour le musée

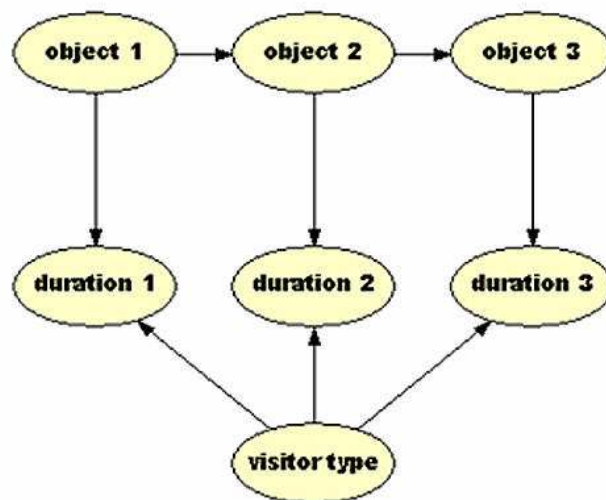


FIGURE 2.9 – Structure du réseau Bayésien appliqué à trois objets

	neutral			Interesting			boring		
	busy	greedy	selective	busy	greedy	selective	busy	greedy	selective
skip	0.2	0.1	0.4	0.1	0.05	0.1	0.35	0.2	0.65
short	0.7	0.1	0.2	0.6	0.05	0.3	0.6	0.2	0.15
long	0.1	0.8	0.4	0.3	0.9	0.6	0.05	0.6	0.2

(a) Table de probabilités conditionnelles des nœuds de durée

P(O1)		P(Oj Oi)	neutral	interesting	boring
Neutral	0.333	neutral	0.6	0.2	0.2
interesting	0.333	interesting	0.2	0.6	0.2
Boring	0.333	boring	0.2	0.2	0.6

(b) Probabilités de transition entre objets

FIGURE 2.10 – Probabilités du réseau Bayésien

dans ces niveaux, des méthodes de classification ont été utilisées (K-means et recherche de seuil entre court et long).

Les probabilités utilisées dans ce réseau (voir figure 2.10) ont été définies par des experts du domaine (conservateurs de musée) et affinées par des observations effectuées sur 50 visiteurs. Les probabilités du premier objet sont fixées à un tiers pour chaque variable. Pour les autres objets, les probabilités de transition de la figure 2.10b sont utilisées. Pour les nœuds de durée, c'est la table de probabilités conditionnelles de la figure 2.10a qui est utilisée.

Ce sont les durées de passage du visiteur obtenues grâce aux capteurs du musée qui servent de données d'entrée au réseau. Le calcul de propagation des probabilités au sein du réseau permet d'obtenir en sortie un profil pour l'utilisateur (plus forte probabilité).

Dans le système proposé par F. Sparacino, ce réseau Bayésien est ensuite étendu pour gérer la sélection de contenu à présenter au visiteur.

Chapitre 3

Principes

3.1 Vue générale

Pour rappeler nos objectifs, nous cherchons à mettre en place une architecture permettant de générer la présentation d'un environnement structuré. Nous proposerons trois approches : interactive, directive et pro-active. En terme d'architecture, nous utilisons trois modules (voir figure 3.1) qui communiquent entre eux par l'intermédiaire d'un serveur :

- Un module de planification, qui produit une suite d'instructions dont l'exécution contrôle la visite.
- Un module de visualisation, qui exécute la visite et constitue l'interface avec l'utilisateur (restitution visuelle/sonore et détection des actions du visiteur).
- Un module d'identification des intentions du visiteur (pour l'approche pro-active), afin de lui proposer une visite correspondant à son profil.

Cette architecture permet de totalement dissocier les différents mécanismes du système afin d'offrir une liberté dans les choix de développement à faire. Cela pourrait être particulièrement intéressant dans le cas du module de planification. À terme, ce choix d'architecture permet également de faciliter la distribution du programme.

3.2 Module de visualisation

3.2.1 Présentation

Le module de visualisation a pour but de permettre à l'utilisateur de naviguer dans le corps humain, à la façon d'un musée virtuel. L'utilisateur peut naviguer de manière libre s'il le souhaite, sans avoir d'explication, ou en mode guidé qui lui permet de suivre un plan proposé par le module de planification d'explications. C'est ce second mode qui nous intéresse plus particulièrement. Les plans proposés par le planificateur sont composés d'instructions de commande que peut comprendre le module de visualisation. Il s'agit d'une panoplie de commandes qui gèrent les déplacements de l'utilisateur et l'affichage d'explications. Afin de pouvoir cerner les objectifs de l'utilisateur, des capteurs sont ajoutés pour le situer dans la scène et détecter s'il interagit avec un objet présenté (sélection d'un organe, passage à proximité d'un organe,...).

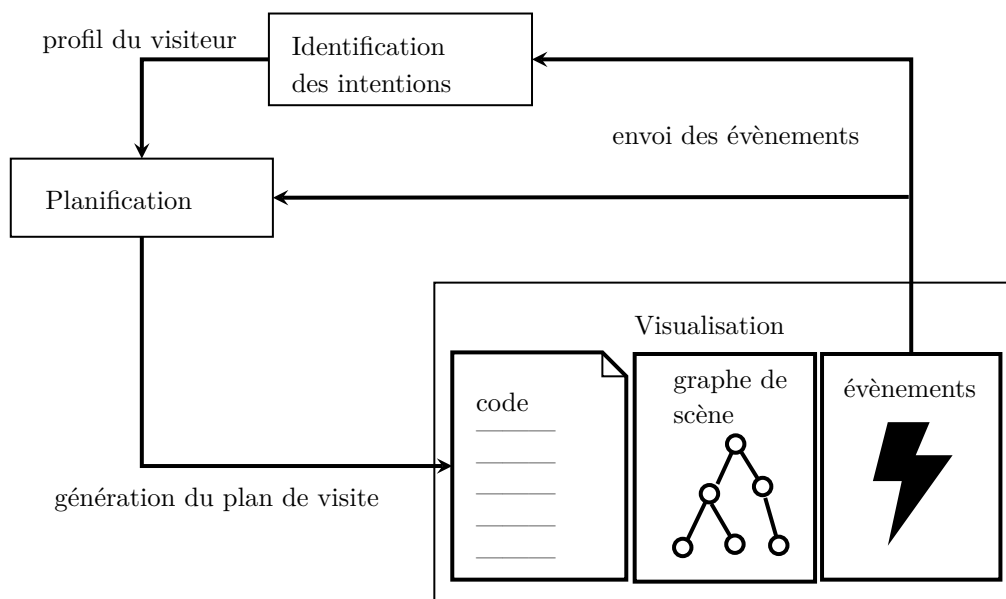


FIGURE 3.1 – Schéma synoptique de l'architecture visualisation/planification

3.2.2 Graphe de scène

Le programme considéré a pour objet la présentation visuelle et sonore du corps humain. Celui-ci est implémenté sous la forme d'un environnement 3D représentant les différents organes du corps. Cette représentation est hiérarchique :

- La racine représente le corps humain.
- Chaque feuille représente un organe.
- Chaque autre nœud intermédiaire représente un groupe d'objets.

Ces nœuds peuvent se voir associer des informations supplémentaires comme :

- des documents multimédia
- des transformations géométriques
- des informations sur leur visibilité (affiché, masqué, transparent)

Il est également possible d'ajouter d'autres nœuds qui ne seront utilisés que si nécessaire (par exemple des flèches ou des capteurs).

3.2.3 Instructions de visite

Afin de pouvoir guider l'utilisateur au sein de l'environnement virtuel du corps humain, différentes solutions étaient possibles. Par rapport à l'architecture choisie, de développement en différents modules, le choix le plus pertinent était l'utilisation d'un système d'instructions de visite. Une liste d'instructions est exécutée par un interpréteur. Elles ont pour effet de modifier le graphe de scène afin de déplacer l'utilisateur ou de présenter des explications. Au fur et à mesure des tests effectués, nous en sommes arrivé à l'utilisation de 10 commandes différentes :

Action	Détails
Filtrage	Masquage d'objets
Déplacement dans l'environnement	Déplacement de l'utilisateur aux coordonnées spécifiées
Déplacement du regard	Déplacement de la caméra pour regarder les coordonnées spécifiées
Mise en avant d'un objet	Affichage des autres objets en transparent
Image	Affichage d'une image à côté de l'objet concerné
Son	Lecture d'un son
Flèche	Fléchage d'un objet pour présenter une partie de celui-ci
Marquage temporel d'un son	Action (décrite à l'aide des commandes) à effectuer à un temps donné d'un son
Texte	Affichage d'un texte
Attente	Pause du guide pendant la présentation d'une explication

3.2.4 Réaction à des évènements

En plus d'afficher les différents objets et de guider l'utilisateur, le module de visualisation doit permettre de récupérer des données pour cerner le comportement de l'utilisateur. Pour cela, des capteurs vont être ajoutés à la scène comme ce qui est proposé dans VRML. Nous avons considéré que deux types de capteurs VRML pouvaient permettre de cerner le comportement de l'utilisateur :

- les capteurs de proximité (ou ProximitySensor), qui permettent de détecter la présence de l'utilisateur dans une zone de la scène.
- les capteurs de toucher (ou TouchSensor), qui permettent de détecter un clic de l'utilisateur sur un objet de la scène.

Le module de planification gère la visite, donc la seule tâche du module de visualisation à ce niveau est de tenir le planificateur au courant des actions de l'utilisateur. Pour les capteurs de proximité, cela revient à indiquer toute entrée et sortie d'une zone concernée. Pour les capteurs de toucher, il s'agit d'indiquer tout clic fait par l'utilisateur sur un objet considéré comme sensible. C'est en fonction de ces évènements que le planificateur peut cerner le comportement de l'utilisateur et ainsi modifier le plan de visite. Le module de visualisation est tenu au courant de ces changements pour ainsi poursuivre avec un nouveau plan de visite.

3.3 Différents guides

3.3.1 Guide interactif

Une première idée pour la planification d'explications a été de ne pas définir de plan pour la visite mais au contraire de laisser l'utilisateur dicter ses intentions. Pour cela, il est possible d'utiliser les capteurs présentés dans la section 3.2.4. Le module de visualisation déclenche un évènement lorsque l'utilisateur interagit avec un capteur et cet évènement est alors envoyé au module de planification qui génère un nouveau plan de visite en cohérence avec les actions réalisées. Ce plan est envoyé au guide du module de visualisation qui l'exécute.

Ainsi, lorsque l'utilisateur entre dans la zone d'un capteur de proximité associé à un organe, les explications de celui-ci lui sont présentées. Elles sont stoppées dès qu'il ressort de cette zone. Dans ce cas, un nouvel évènement est envoyé au planificateur qui génère un nouveau plan.

L'utilisateur peut également cliquer sur un organe qu'il souhaite étudier. Dans ce cas, il sera guidé jusqu'à la zone du capteur de proximité de cet organe et comme précédemment, les explications lui seront ainsi présentées. Si l'utilisateur revient sur un organe déjà vu en partie, seules les explications qu'il n'a pas encore eu lui sont données. Pour cela, chaque explication doit avoir un marqueur qui indique si elle a été présentée ou non.

L'avantage de cette méthode est que l'utilisateur a un contrôle total sur ce qu'il va voir. Il ne devrait donc pas s'ennuyer lors de la visite. Mais l'inconvénient est justement que le guide ne contrôle rien du tout. Cela implique qu'en utilisant cette méthode, il n'est pas sûr, tout d'abord, que l'utilisateur ait bien compris la fonction d'un organe (ou sous-organe) au sein de l'organisme, et ensuite, que les différentes notions indispensables lui aient été présentées.

3.3.2 Guide directif

Une deuxième approche consiste à laisser le contrôle de la visite au guide. Pour cela, on peut envisager de laisser le soin à un expert de spécifier une visite en donnant une liste d'organes à présenter. Une autre solution consiste à utiliser la structure du système à présenter afin de guider la visite. Dans le cas du corps humain, la description peut se faire selon plusieurs niveaux (voir figure 3.2) :

- le corps
- les systèmes fonctionnels (ex : respiratoire, digestif)
- les organes
- les parties de ces organes (que nous appellerons sous-organes)

Une visite consiste alors en un parcours de ces hiérarchies à l'image du code suivant :

```
#Presentation d'un systeme du corps avec parcours du graphe en largeur  
presenterExplications(systeme)  
for organe in systeme:  
    presenterExplications(organe)  
for organe in systeme:  
    for sousorgane in organe:  
        presenterExplications(sousorgane)
```

```
#Presentation d'un systeme du corps avec parcours du graphe en profondeur  
presenterExplications(systeme)  
for organe in systeme:  
    presenterExplications(organe)  
    for sousorgane in organe:  
        presenterExplications(sousorgane)
```

```
#Presentation d'un organe  
presenterExplications(systeme)  
presenterExplications(organe)  
for sousorgane in organe:  
    presenterExplications(sousorgane)
```

```
#Presentation d'un sous-organe  
presenterExplications(systeme)  
presenterExplications(organe)  
presenterExplications(sousorgane)
```

Nous pouvons remarquer qu'il est possible de parcourir le graphe en profondeur ou en largeur pour la présentation d'un système du corps humain. Les deux solutions ont été implémentées mais la première est bien sûr la plus intuitive : présenter le système, présenter le premier organe et ses sous-organes, le deuxième organe et ses sous-organes,...

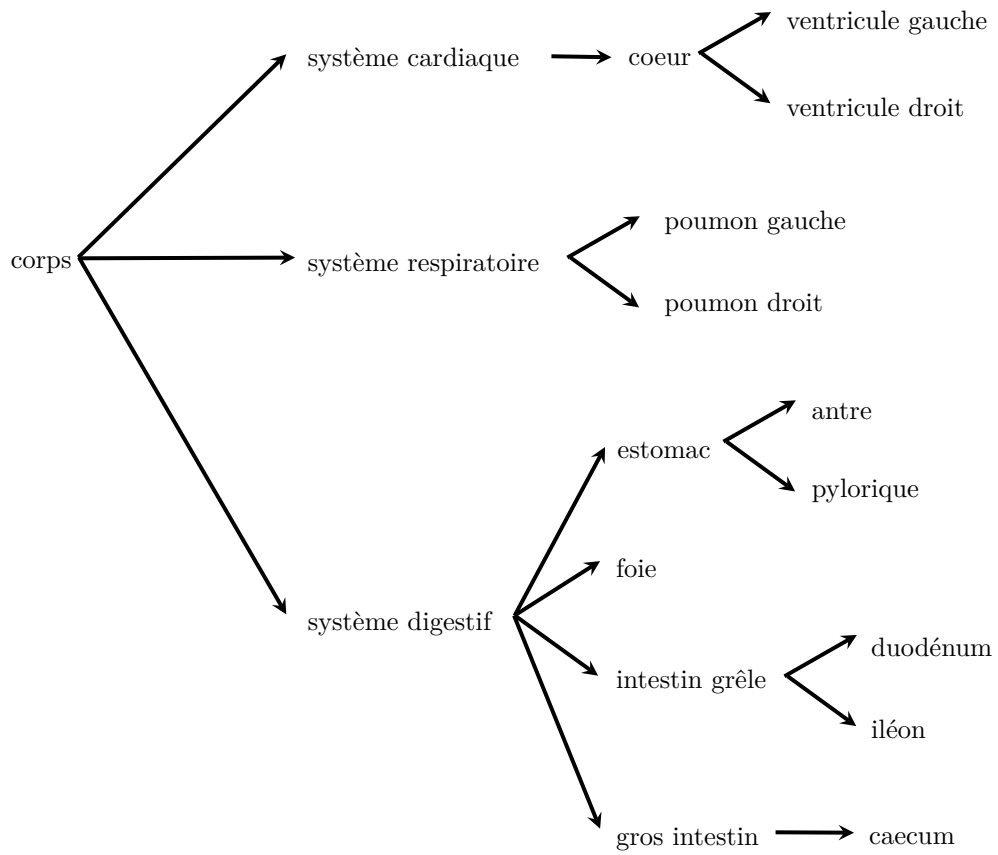


FIGURE 3.2 – Exemple de graphe de description du corps humain

Du graphe aux instructions

Afin de générer un plan de visite, il est nécessaire de passer du parcours du graphe à une suite d'instructions.

Pour la présentation d'un système, nous commençons par filtrer les objets pour n'afficher que les organes appartenant à ce système. Nous pouvons ensuite commencer à présenter au visiteur les différentes explications liées à ce système en utilisant les instructions propres à chaque média (son, image, texte).

Nous passons ensuite à la présentation d'un organe. Celui-ci est donc mis en avant par rapport aux autres du système, en mettant les autres en transparence. L'utilisateur est ensuite déplacé dans l'environnement jusqu'à arriver devant l'organe et son regard est dirigé vers celui-ci. C'est à ce moment-là que les explications de cet organe sont présentées au visiteur.

Pour finir, nous présentons les sous-organes. Comme ceux-ci font parti du modèle 3D de l'organe, il a fallu trouver un moyen de les mettre en avant. Le plus simple est d'ajouter une flèche à la scène pour pointer la zone du sous-organe. Pour la présentation d'un sous-organe, nous commençons donc par le flécher avant de présenter les différentes explications qui lui sont associées.

La description du corps humain peut également se faire en terme de circulation de matières et de fluides (sang, nourriture,...) et de leur transformation. Par exemple, nous pouvons définir que le passage de la nourriture se fait dans l'ordre estomac \rightarrow intestin grêle \rightarrow gros intestin. Ainsi, si l'utilisateur demande la présentation du passage de la nourriture, ces différents organes lui seront présentés dans cet ordre.

3.3.3 Guide pro-actif

Les deux guides présentés au-dessus permettent de répondre directement à une demande de l'utilisateur mais sans proposer un plan complet de visite.

Identification des intentions de l'utilisateur

Afin de comprendre l'utilisateur, le planificateur utilise les données des capteurs du module de visualisation à partir desquelles il en ressort un profil d'utilisateur. Nous utilisons la même classification des utilisateurs que celle retenue par Flavia Sparacino (voir section 2.6.2).

Nous utilisons également un réseau Bayésien pour notre guide. Par contre, comme nous ne disposons pas des mêmes moyens que F. Sparacino (experts du domaine, visiteurs pour expérimentation), j'ai décidé d'utiliser les mêmes valeurs de probabilités que celles qu'elle utilise pour faire fonctionner le réseau Bayésien propre à notre outil. Cela a permis de valider l'intérêt de cette solution, dont les valeurs pourront ensuite être affinées par expérimentation. Par contre, nous n'utilisons que la partie principale du réseau Bayésien proposée par Sparacino, nous n'avons pas retenu l'extension du réseau qui s'occupe de la sélection du contenu à présenter au visiteur. En effet, cette dernière ne gère pas l'ordre des explications aussi bien que ce dont nous aurions besoin, dans le cadre de données pédagogiques.

En entrée de notre réseau, nous utilisons donc les valeurs de durée obtenues à l'aide des capteurs de proximité de l'environnement. Comme expliqué pour le réseau de F. Sparacino, c'est la propagation des probabilités qui permet d'obtenir en sortie des valeurs pour chaque type de profil (busy, selective, greedy). Nous assignons donc à l'utilisateur le profil qui obtient la plus forte probabilité.

Graphes d'explication

À la manière de ce que proposent Veyret et al. (voir section 2.5), nous utilisons des graphes pour l'ordonnancement des explications. Dans leurs travaux, les objets présentés (des poissons) sont dynamiques et ont un comportement peu prévisible, mais le visiteur est statique. Dans notre cas, c'est l'inverse. Les objets sont statiques, mais l'utilisateur peut quant à lui se déplacer dans l'environnement. C'est donc à ce niveau que nos deux travaux se rapprochent, il s'agit dans les deux cas de gérer des déplacements peu prévisibles.

Les graphes seront donc un point clé de notre système à guide pro-actif. Comme nous l'avons vu dans la section précédente, nous utilisons un réseau Bayésien pour identifier le profil de l'utilisateur parmi les trois types busy, selective et greedy. Nous avons donc imaginé la création de trois graphes par objet : un graphe pour chaque profil. Chaque objet possède une liste d'explications.

Le graphe pour le profil greedy contient toutes les explications de l'objet.

Le graphe pour le profil busy contient seulement les explications les plus importantes de l'objet.

Le graphe pour le profil selective est un compromis entre les deux graphes précédents.

Lors de la présentation d'un objet, un des trois graphes sera choisi selon le profil actuel de l'utilisateur. Si le profil change en cours de visite, la visite est replanifiée en choisissant le graphe correspondant.

Génération du plan de visite

Le planificateur d'explications est capable de générer des plans de visite de manière semi-autonome. En effet, il adapte des plans de base, qui ont été établis au préalable par des experts du domaine étudié, afin de proposer une visite qui correspond aux objectifs de l'utilisateur. Chaque plan est une liste des objets à présenter durant la visite. Pour par exemple présenter le cœur, on peut établir le plan suivant :

```
coeur  
oreillette  
ventricule
```

Au démarrage de l'application, l'utilisateur choisit donc une des visites définies par les experts. Par la suite, chacun des objets est alors présenté en utilisant les graphes liés au profil de l'utilisateur. Si le profil de l'utilisateur change en cours de présentation, le plan de visite est alors modifié en utilisant les graphes pour ce profil, que l'on adapte pour ne pas présenter des explications déjà proposées.

Chapitre 4

Mise en œuvre

4.1 Visualisation

4.1.1 Commandes de visite

En suivant le principe présenté dans la section 3.2.3, nous avons implémenté 10 commandes différentes pour gérer les déplacements dans l'environnement et la présentation des explications. Le programme de visite est enregistré dans un simple fichier texte où chaque ligne contient une commande (voir figure 4.1). Voici donc les 10 commandes qui sont actuellement utilisables pour une visite :

Action	Commande
Filtrage	filtrer objet1,objet2,...
Déplacement dans l'environnement	allerEn x,y,z
Déplacement du regard	regarderEn x,y,z
Mise en avant d'un objet	avant objet
Image	image objet,fichier,larg,haut
Son	son fichier
Flèche	fleche x,y,z,angle
Marquage temporel d'un son	breakpoint Xsec
Texte	texte txt
Attente	pause Xsec

La fonction d'analyse syntaxique utilisée pour lire le fichier de plan de visite a été conçue de manière à pouvoir ajouter très facilement de nouvelles commandes en cas de besoin. Ceci a été pratique lors du développement du module de visualisation, pendant lequel nous avons ainsi pu ajouter des commandes auxquelles nous n'avions pas pensé au début.

4.1.2 Environnement

Afin de simplifier la visualisation du corps humain, il a été fait le choix de développer un environnement en 2,5D, c'est à dire à mi-chemin entre la 2D et la 3D. Ainsi, les organes apparaissent sous la forme d'images plaquées sur des planches invisibles placées le long de l'axe z.

Afin d'éviter trop de travail répétitif de mise en place, nous avons fait le choix d'une scène simple de quelques organes. Cela ne change en rien le fait que les solutions mises en œuvre peuvent être appliquées à des systèmes complexes.

```
avant coeur
allerEn 0.0,1.75,0.94
regarderEn 0.0,1.75,-0.06
son coeur_gauche.ogg
breakpoint 2.5
fleche -0.16,1.75,0.55,-90
breakpoint 3.5
fleche 0.0,1.7,0.55,90
pause 0.5
avant coeur
allerEn 0.0,1.75,0.94
regarderEn 0.0,1.75,-0.06
fleche -0.16,1.75,0.55,-90
pause 1.0
son ventr_ga.ogg
pause 0.5
avant coeur
allerEn 0.0,1.75,0.94
regarderEn 0.0,1.75,-0.06
fleche 0.0,1.7,0.55,90
pause 1.0
son ventr_dr.ogg
pause 0.5
```

FIGURE 4.1 – Exemple de fichier de plan de visite (ici, pour présenter le coeur)

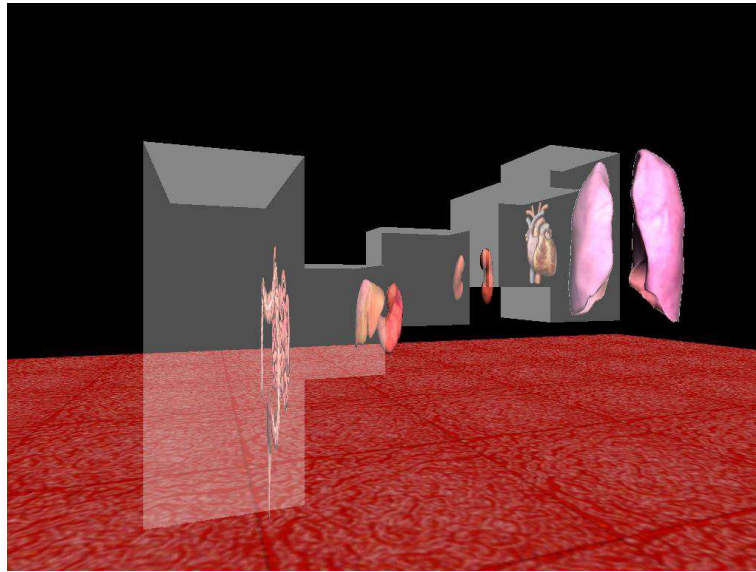


FIGURE 4.2 – Zones des capteurs de proximité (affichés en transparence) pour certains organes

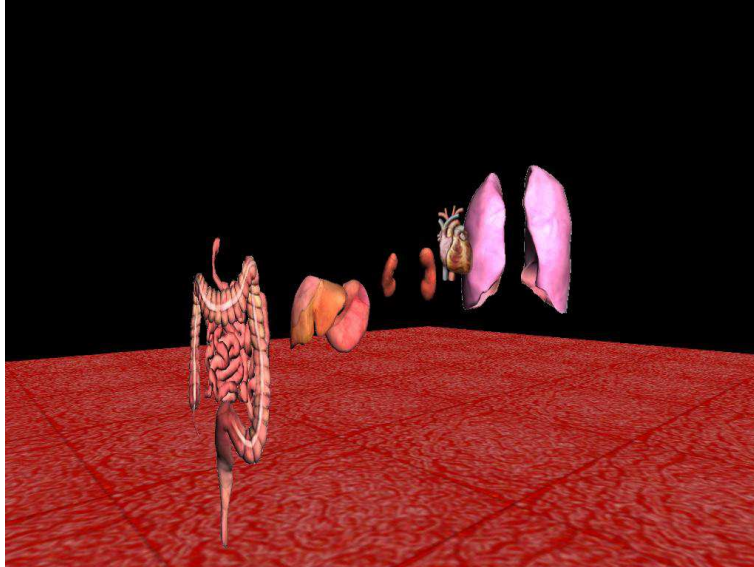
4.1.3 Capteurs

Capteur de proximité

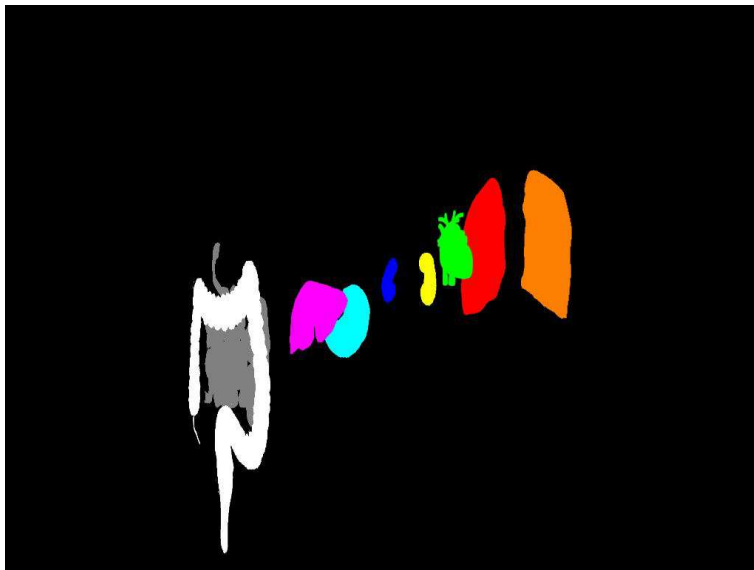
Pour ce type de capteur, je me suis en partie reporté à la définition du nœud de capteur de proximité pour VRML [21]. Il s'agit donc d'une boîte rectangulaire pour laquelle on définit un centre et une taille suivant les coordonnées x,y,z . Le capteur génère des évènements lorsque l'utilisateur y entre et en sort pour en informer le module de planification. Nous ne nous intéressons pas aux mouvements de l'utilisateur une fois qu'il est dans le capteur. Il s'agit d'une simplification par rapport aux capteurs VRML. Nous partons du fait que si l'utilisateur est à l'intérieur de la zone du capteur, cela signifie qu'il regarde l'objet concerné.

Capteur de toucher

Afin de déterminer les objets sur lesquels l'utilisateur clique, nous avons mis en œuvre une technique appelée "fausse couleur" qui consiste à définir une carte des éléments cliquables. Dans celle-ci, une couleur particulière est appliquée à chaque élément. Ainsi, lorsque l'utilisateur clique dans l'environnement, un évènement récupère les coordonnées du pixel cliqué et il suffit donc de lire la carte de fausses couleurs pour récupérer la couleur de ce pixel. Comme chaque objet possède sa propre couleur, on en déduit ainsi l'objet cliqué.



(a) Environnement réel



(b) Même environnement en fausses couleurs

FIGURE 4.3 – Technique de la "fausse couleur"

4.2 Planification d'explications

4.2.1 Graphe de représentation du corps humain

Pour le guide directif, nous étions parti sur le langage Prolog. Nous avons donc défini des prédicats pour le graphe de description du corps humain, pour les annotations et un prédicat **presenter(Partie)** qui permet de générer un plan de visite en fonction des autres prédicats.

Prédicats pour le graphe de description du corps humain :

Prédicat	Description
systeme(Sys)	déclaration d'un système du corps humain
organe(Sys,Org,X,Y,Z,Larg,Haut)	déclaration d'un organe de Sys
sousorgane(Org, Sorg, X, Y, Z)	déclaration d'un sous-organe de Org

Prédicats pour les annotations (liées à un objet Part) :

Prédicat	Description
image(Fic, Part, Larg, Haut)	affichage d'une image
son(Fic, Part)	lecture d'un son
texte(Txt, Part)	affichage d'un son

4.2.2 Réseau Bayésien

Pour le guide pro-actif, le module de planification a été développé en Python. Nous avons donc utilisé un module Python pour la création du réseau Bayésien, PyAgrum¹. Il s'agit d'un wrapper (surcouche) pour la librairie aGrUM développée en C++.

4.3 Interaction visualisation/planification

4.3.1 Communication par socket

Comme expliqué dans la section 3.1, nous avons fait le choix d'une architecture modulaire. Afin de donner encore plus de liberté pour le développement des différents modules, ceux-ci communiquent entre eux par socket. Ainsi, d'une part, les modules peuvent être développés dans des langages différents, et d'autre part, ils peuvent s'exécuter sur des machines différentes reliées en réseau. Dans notre choix de guide pro-actif, les deux modules (visualisation et planification) sont écrits en Python. Mais dans nos essais précédents, nous avons écrit le module de planification en langage Prolog.

Ce choix permet donc de tester et comparer plus facilement différentes solutions sans avoir à s'inquiéter de l'interfacage entre les différents modules. Il suffit juste de pouvoir implémenter une communication par socket dans chacun des modules.

4.3.2 Langage de communication

Pour permettre la communication entre les deux modules, de visualisation et de planification, un langage simple a été mis en place, à l'image des instructions de commande pour la visite (voir section 4.1.1). Dans l'implémentation actuelle, c'est le module de visualisation qui contrôle principalement le dialogue. Il envoie donc des messages pour avertir le planificateur de nouveaux

1. aGrUM/PyAgrum : <http://agrum.lip6.fr/doku.php?id=pyagrum:index>

événements ou pour lui indiquer ce qu'il doit faire. Voici les différents messages que peut envoyer le module de visualisation :

ProxSens IN nom t : Indique que l'utilisateur est entré dans la zone du capteur de proximité 'nom' au temps 't'.

ProxSens OUT nom t : Indique que l'utilisateur est sorti de la zone du capteur de proximité 'nom' au temps 't'.

TouchSens nom : Indique que l'utilisateur a cliqué sur le capteur de toucher 'nom'.

start : Demande au planificateur de générer un plan de visite.

update : Demande au planificateur de mettre à jour le réseau Bayésien en fonction des informations de capteurs.

Pour les trois premiers messages, le module de planification se contente de répondre par un message de validation pour indiquer qu'il a bien pris en compte l'information. Pour le message "start", le planificateur renvoie un message de validation lorsqu'il a généré un plan de visite. Ainsi, le module de visualisation n'a plus qu'à lire le fichier de plan pour guider l'utilisateur. Pour le message "update", le planificateur renvoie un simple message de validation lorsque le profil d'utilisateur obtenu par le réseau Bayésien ne change pas. Dans le cas inverse, le planificateur génère un nouveau plan de visite en fonction du profil obtenu et en informe le module de visualisation pour qu'il lise de nouveau le fichier de plan.

Chapitre 5

Conclusion

Notre travail a permis de faire les choix essentiels à la mise en œuvre de notre atlas interactif du corps humain par rapport aux objectifs à atteindre.

Dans un premier temps, nous avons mis en place un module de visualisation qui permet à l'utilisateur de se déplacer dans un environnement 3D représentant les organes du corps humain. C'est dans cette partie que sont gérés les principes de filtrage des objets, d'affichage des annotations et de capture des actions de l'utilisateur. Ainsi, l'utilisateur ne voit que les organes concernés par le thème de la visite et le guide lui présente des explications qui peuvent être de différentes formes. Dans l'état actuel de notre outil, il peut s'agir d'images, de sons et de textes. La mise en place de différents stimuli va dans le sens des travaux de recherche effectués par Barraclough et Guymer [22] sur le fonctionnement de l'apprentissage par l'Homme dans lesquels ils ont démontré que l'utilisation de différents sens permet d'acquérir plus d'informations. En ce qui concerne les actions de l'utilisateur, on peut le situer dans la scène et chronométrer le temps qu'il reste devant chaque objet grâce à des capteurs de proximité, et on peut détecter les clics qu'il effectue sur les objets à l'aide de capteurs de toucher. Ces actions génèrent des événements qui sont transmis au module de planification pour qu'il en déduise les intentions de l'utilisateur et génère un plan de visite en cohérence avec ces données. Ce plan de visite est composé de commandes qui permettent le déplacement de l'utilisateur dans l'environnement et la présentation des explications.

Nous avons testé différentes approches pour le planificateur. Une première a été la mise en place d'un guide que l'on peut considérer comme interactif en suivant la classification proposée par Veyret [12]. En effet, celui-ci génère un plan de visite qui résulte directement d'une action de l'utilisateur (entrée dans une zone de capteur de proximité ou clic sur un objet). L'avantage de cette méthode pour l'utilisateur est qu'il est le maître de la visite. D'un point de vue pédagogique, c'est un inconvénient car on ne peut pas s'assurer d'avoir présenté les différentes notions essentielles du système à étudier. Un autre inconvénient est que l'utilisateur peut demander des explications sans suivre un schéma de visite précis. De ce fait, il peut ne pas comprendre la fonction des objets au sein du système étudié.

Une seconde approche pour corriger ce dernier défaut a été de proposer un guide plus directif que le précédent. Celui-ci utilise un graphe de représentation du corps humain pour ne plus présenter simplement un objet mais également sa hiérarchie. Cela permet à l'utilisateur de situer à chaque fois la fonction d'un objet au sein du système étudié.

Nous avons ensuite réfléchi aux principes d'un dernier guide, pro-actif, qui permet de répon-

dre aux besoins pédagogiques de l'outil. Celui-ci se base sur l'utilisation de graphes d'explications définis par un expert du domaine. Ces graphes permettent ainsi d'aborder les différentes notions liées à un objet dans un ordre logique pour permettre à l'utilisateur de bien tout comprendre. Chaque explication peut posséder une indication sur son importance afin de décider du moment à partir duquel on peut arrêter de décrire un objet. Lorsqu'on commence à présenter un objet, l'utilisateur est bloqué sur celui-ci tant que toutes les explications dites importantes ne sont pas présentées. Enfin, pour adapter le "discours" du guide aux intentions de l'utilisateur, un réseau Bayésien nous permet de déduire le profil de l'utilisateur (busy, selective, greedy) en fonction du temps resté devant chaque objet. Ainsi, pour présenter un objet, le guide choisira le graphe correspondant au profil de l'utilisateur.

Durant ce stage, nous avons dû réfléchir aux différents principes à mettre en œuvre pour répondre aux objectifs d'un atlas interactif du corps humain. La dernière solution, de guide pro-actif, nous semble appropriée, mais il faudrait continuer son développement pour pouvoir l'expérimenter et ainsi en tirer des conclusions. Pour cela, il serait nécessaire d'implémenter le moteur d'explications, qui permet de passer d'un graphe à un autre, et il faudrait également faire appel à des experts du domaine anatomique pour mettre en œuvre des graphes d'explications plus complet que ceux que j'ai pu créer à partir de mon peu de connaissances du domaine. Pour ensuite rendre l'application distribuable, il pourrait être intéressant de suivre les perspectives que nous proposons dans le chapitre suivant.

Chapitre 6

Perspectives

6.1 Utilisation de modèles 3D

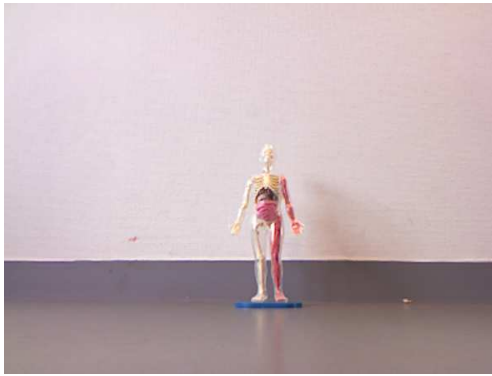
Afin de ne pas perdre de temps sur la modélisation du corps humain, nous avons utilisé des images d'organes. Pour rendre l'environnement plus réaliste, il faudrait utiliser des modèles 3D des organes. Pour cela, plusieurs moyens peuvent être utilisés. Nous allons en détailler deux.

6.1.1 Kinect

La Kinect est un périphérique de la console Xbox 360 fabriquée par Microsoft. Son succès est dû à son faible coût comparé aux équipements équivalents qui étaient sur le marché avant sa sortie. Elle possède un capteur de couleur et un de profondeur.

Nous avons à notre disposition un mannequin d'une trentaine de centimètres qui contient les organes principaux du corps humain. Nous avons donc envisagé de scanner cet objet sous différents angles à l'aide de la Kinect pour en faire ressortir des modèles 3D après traitement. C'est un travail qui a été effectué par de nombreuses personnes dont on peut facilement retrouver des vidéos sur Internet [23].

Malheureusement, la résolution du capteur de profondeur de la Kinect est trop faible pour pouvoir effectuer cette manipulation avec un objet aussi petit (voir figure 6.1).



(a) Capture de la couleur



(b) Capture de la profondeur

FIGURE 6.1 – Capture de notre mannequin avec la Kinect

6.1.2 Modélisation du Visible Human Project

Une autre solution est d'utiliser les photographies effectuées dans le cadre du Visible Human Project [24]. Ce projet a été conçu entre 1989 et 1995. Il consistait à photographier des tranches de corps humain obtenues sur des cadavres des deux sexes (1871 tranches de 1mm pour le cadavre masculin).



FIGURE 6.2 – Coupe au niveau de l'abdomen issue du Visible Human Project

L'article d'Harry Brenton et al. [2] décrit les différentes étapes pour la modélisation d'objets 3D à partir d'images 2D, qui sont les suivantes (voir figure 6.3) :

1. Numérisation et amélioration de la qualité : cette étape consiste à avoir des tranches de toute la longueur de l'objet à modéliser. C'est ici que nous pouvons utiliser les photographies du Visible Human Project. Les images sont ensuite traitées, en effectuant par exemple de la réduction de bruit, pour en améliorer la qualité.
2. Reconstruction : il faut segmenter les images en traçant sur chacune la zone à modéliser. On obtient ainsi un objet 3D à partir des images.
3. Simplification du mesh ("réseau de mailles") : l'objet 3D est traité pour le simplifier.
4. Rendu de surface.
5. Application d'une texture.

Ces différentes étapes sont fastidieuses et pourront être menées ultérieurement comme il ne s'agit pas de l'objectif principal de nos travaux.

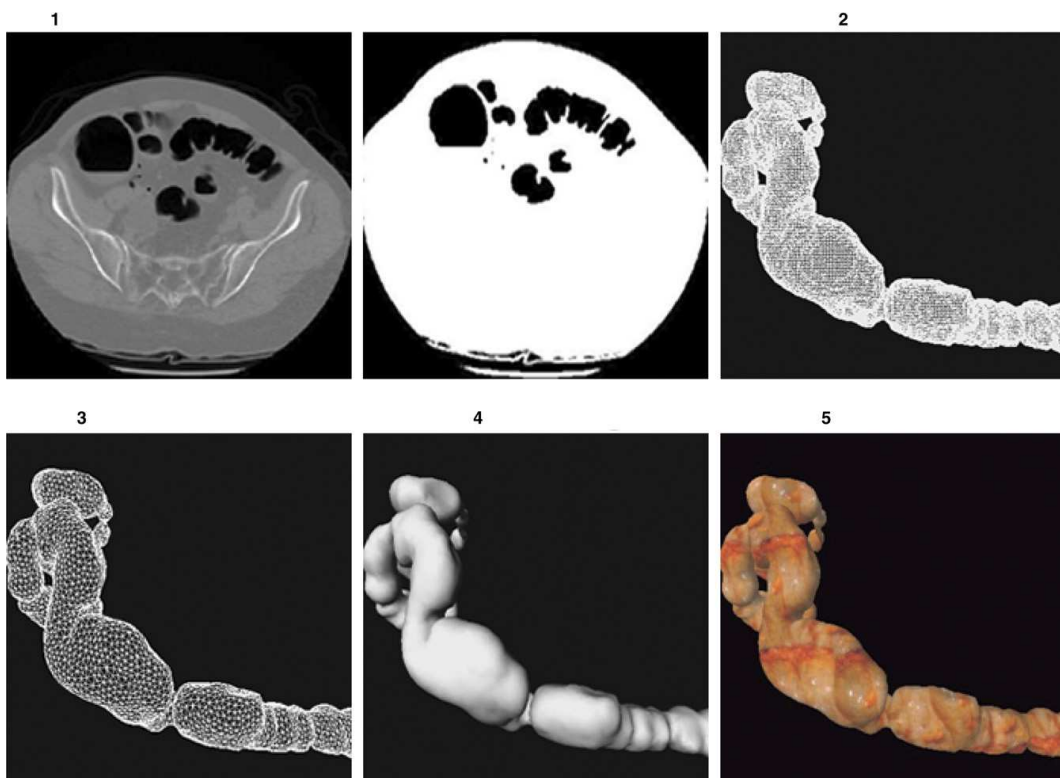


FIGURE 6.3 – Les différentes étapes de modélisation 3D

6.2 Affichage des annotations

Si l'utilisation de modèles 3D est envisagée, il faudra également se reporter aux travaux présentés dans la section 2.4 pour gérer l'affichage des annotations 3D. Il s'agit d'un travail annexe qui permettrait d'améliorer la qualité de l'atlas interactif du corps humain.

6.3 Amélioration des capteurs de proximité

Les zones de nos capteurs de proximité sont actuellement de simples boîtes rectangulaires (voir section 4.1.3). Il pourrait être intéressant de s'inspirer des travaux de Benford et Fahlén [7] et Julier [6] sur le modèle d'interaction spatiale (voir section 2.3.2). Ainsi, la zone du capteur ne serait plus directement liée à la géométrie de l'objet concerné mais à d'autres critères, comme par exemple l'importance de l'objet par rapport au thème abordé.

6.4 Expérimentation

Le travail de réflexion et de mise en œuvre du projet ne nous a pas laissé le temps de mettre en place des séances d'expérimentations. Il pourrait être intéressant d'en faire pour plusieurs raisons.

Premièrement, cela nous permettrait d'affiner les données de probabilités utilisées dans le réseau Bayésien. En effet, comme nous l'avons expliqué précédemment, nous avons utilisé les mêmes valeurs que F. Sparacino dans son article.

Dans un second temps, d'autres expérimentations permettraient d'évaluer l'intérêt de notre outil au niveau pédagogique pour les utilisateurs.

Bibliographie

- [1] Luca Chittaro and Roberto Ranon. Web3d technologies in learning, education and training : motivations, issues, opportunities. *Computers and Education*, 49 :3–18.
- [2] Harry Brenton, Juan Hernandez, Fernando Bello, Paul Strutton, Sanjay Purkayastha, Tony Firth, and Ara Darzi. Using multimedia and web3d to enhance anatomy teaching. *Comput. Educ.*, 49 :32–53, August 2007.
- [3] U.S National Library of Medicine. Unified Medical Language System. <http://www.nlm.nih.gov/research/umls/>. [Online ; accessed 01-June-2011].
- [4] The University of Washington Structural Informatics Group. Foundational Model of Anatomy. <http://sig.biostr.washington.edu/projects/fm/AboutFM.html>. [Online ; accessed 01-June-2011].
- [5] Craig W. Reynolds. Flocks, herds, and schools : A distributed behavioral model. In *SIGGRAPH '87 : Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 25–34, New York, NY, USA, 1987. ACM.
- [6] Simon Julier, Marco Lanzagorta, Yohan Baillet, and Dennis Brown. Information filtering for mobile augmented reality, 2002.
- [7] Steve Benford and Lennart Fahlén. A spatial model of interaction in large virtual environments, 1993.
- [8] Simon Julier Marco, Marco Lanzagorta, Yohan Baillet, and Lawrence Rosenblum. Information filtering for mobile augmented reality. In *In Proc. ISAR 2000*, pages 3–11, 2000.
- [9] D. Lenne, I. Thouvenin, and S. Aubry. Supporting design with 3d-annotations in a collaborative virtual environment. *Research in Engineering Design*, 20(3) :149–155, september 2009.
- [10] Stéphane Aubry. *Annotations et gestion des connaissances en environnement virtuel collaboratif*. PhD thesis, Université de Technologie de Compiègne, 2007.
- [11] Blaine Bell, Steven Feiner, and Tobias Höllerer. View management for virtual and augmented reality. pages 101–110. ACM Press, 2001.
- [12] Morgan Veyret. *Un guide virtuel autonome pour la description d'un environnement réel dynamique. – Interaction entre la perception et la prise de décision*. PhD thesis, Université de Bretagne Occidentale, Laboratoire d'Informatique des Systèmes Complexes – EA3883, March 2009.
- [13] Xiang Yuan and Yam San Chee. Embodied tour guide in an interactive virtual art gallery. In *International Conference on Cyberworlds*, 2003.
- [14] Morgan Veyret, Eric Maisel, and Jacques Tisseau. Guide virtuel autonome immergé dans un environnement réel dynamique. *Technique et Science Informatiques*, January 2009.

- [15] J. Ibanez, R. Aylett, and R. Ruiz-Rodarte. Storytelling in virtual environments from a virtual guide perspective, 2003.
- [16] Flavia Sparacino. The museum wearable : real-time sensor-driven understanding of visitors' interests for personalized visually-augmented museum experiences. In *In : Proceedings of Museums and the Web (MW2002)*, pages 17–20, 2002.
- [17] Flavia Sparacino. Sto(ry)chastics : a bayesian network architecture for user modeling and computational storytelling for interactive spaces. pages 54–72, 2003.
- [18] E. Hooper-Greenhill. *Museums and their visitors*. Routledge, 1999.
- [19] D. Dean. *Museum Exhibition : Theory and Practice*. Routledge, 1994.
- [20] B. Serrell. The question of visitor styles. volume 7.1, pages 48–53. Jacksonville AL : Visitor Studies Association, 1996.
- [21] LightHouse3d. ProximitySensor Node. <http://www.lighthouse3d.com/vrml/tutorial/index.shtml?proxi>. [Online; accessed 30-May-2011].
- [22] A. Barraclough and I. Guymer. Virtual reality – a role in environmental engineering education? *Water Science and Technology*, 38(11) :303–310, 1998.
- [23] DFKI. 3D Reconstruction with Kinect. http://www.youtube.com/watch?v=YH58u_057Ac. [Online; accessed 01-June-2011].
- [24] U.S National Library of Medicine. The Visible Human Project. http://www.nlm.nih.gov/research/visible/visible_human.html. [Online; accessed 31-May-2011].