

Université Jean Monnet de Saint Etienne
Projet Master OIV2
2003-2004

Génération d'animations 3D à partir de modèles de scénarios

Rapport final

Jihène BANNOUR

le 15 Septembre 2004

Responsables de stage : Benoît GEORIS et Monique THONNAT
INRIA - Sophia Antipolis



REMERCIEMENTS

Je tiens tout d'abord à remercier Monique Thonnat pour m'avoir accueilli au sein de l'équipe ORION et pour son soutien pendant le stage.

Je suis très reconnaissante envers Benoît Georis qui m'a encadré tout au long de mon stage et qui a toujours été à mon écoute. Son point de vue complémentaire a été très stimulant.

Un grand merci également à François Brémond pour ses précieux conseils tout au long de mon stage.

Je remercie également Christophe Tornieri pour sa disponibilité et la qualité de ses explications et conseils.

Merci également à Bernard Boulay qui n'a pas hésité à m'aider tout au long de mon stage.

Je tiens à remercier Thinh Vu Van pour les discussions enrichissantes que nous avons eues.

Merci aussi à Catherine Martin, l'assistante du projet ORION pour son aide précieuse au niveau administratif.

Plus généralement, je remercie le reste de l'équipe pour leurs divers conseils, leur disponibilité et leur soutien amical. Sans l'aide de toute l'équipe, ce travail n'aurait pas pu s'accomplir .

RESUME

L'équipe ORION (INRIA Sophia-Antipolis) est une équipe multi-disciplinaire dont les compétences se situent à la frontière de l'intelligence artificielle, du génie logiciel et de la vision artificielle. Une large part de ses activités est consacrée à l'interprétation automatique d'images fixes et de séquences vidéos.

Ce stage se situe dans le cadre des travaux du projet ORION sur l'interprétation automatique de séquences vidéos pour la reconnaissance de comportements humains. L'équipe ORION a conçu une plate-forme d'interprétation vidéo permettant la détection d'objets mobiles, le suivi de leur trajectoire et la reconnaissance de certains comportements prédéfinis par des experts humains dans une base de scénarios. Cette plate-forme a été notamment testée sur des applications de vidéo surveillance dans les métros et les agences bancaires.

L'objectif de mon travail est de générer automatiquement des animations 3D à partir de modèles de scénarios définis par les experts humains. D'une part, ceci permet aux experts de valider leur modélisation de scénarios. D'autre part, des séquences vidéos synthétiques peuvent être générées à partir de ces animations 3D, et servir de référence pour évaluer les algorithmes de vision.

Une animation 3D est générée en deux étapes principales. D'abord, je propose de transformer le modèle de scénario défini par l'expert (information tel que actions, états, contraintes, environnement 3D de la scène) en un modèle de visualisation contenant les informations nécessaires pour une animation 3D. J'appelle cette étape "transformation de scénario". En second lieu, je génère, à partir du modèle de visualisation défini, des instances de scénarios. Il s'agit d'initialiser les différents paramètres requis pour l'animation 3D : initialiser les instants et positions clés, générer des trajectoires, fixer les paramètres de rendu tels que le contraste ou l'éclairage. J'appelle cette étape "instanciation".

Les premiers résultats obtenus sont prometteurs. J'ai généré automatiquement des animations 3D pour deux applications différentes : une agence bancaire et un train.

Mots clés : animation 3D, modèles de scénarios, évaluation d'algorithmes de vision, interprétation vidéo.

Table des matières

1	Introduction	6
1.1	Présentation de l'équipe ORION de l'INRIA	6
1.1.1	Présentation générale de l'INRIA	6
1.1.2	Présentation de l'INRIA de Sophia-Antipolis	6
1.1.3	Présentation de l'équipe ORION	7
1.2	Descriptif du sujet de stage	8
1.3	Objectifs et motivations du stage	8
2	La plate-forme VSIP	9
2.1	Présentation	9
2.2	Extension	11
3	Génération d'animations 3D	12
3.1	Etat de l'art	12
3.2	Contexte de la scène	13
3.2.1	Présentation du contexte	13
3.2.2	Visualisation du contexte	15
3.3	Représentation de scénarios	15
3.3.1	Modèle générique des scénarios	16
3.4	Transformation de scénarios	17
3.4.1	Méthode de transformation de scénarios	17
3.4.2	Exemple de scénario simple	19
3.4.3	Exemple de scénario complexe	20
3.5	Instanciation	21
3.5.1	Positions et instants clés	22
3.5.2	Génération de trajectoires	22
3.5.3	Les paramètres de rendu	23
4	Résultats et discussion	24
5	Conclusion et perspectives	27
5.1	Conclusion	27
5.2	Perspectives	27
6	Annexe	30
6.1	Méthode du Pivot de Gauss implémentée	30
6.2	Article accepté	31

1 Introduction

1.1 Présentation de l'équipe ORION de l'INRIA

Mon stage se déroule dans l'équipe ORION, une des équipes de l'unité de recherche de l'INRIA (Institut National de Recherche en Informatique et en Automatique) de Sophia-Antipolis.

1.1.1 Présentation générale de l'INRIA

Créée en 1967 à Rocquencourt près de Paris, l'INRIA est un établissement public à caractère scientifique et technologique placé sous la double tutelle du ministère de la recherche et du secrétariat d'Etat à l'industrie.

Les principales missions de l'INRIA sont :

- Entreprendre des recherches fondamentales et appliquées.
- Réaliser des systèmes expérimentaux.
- Organiser des échanges scientifiques internationaux.
- Assurer le transfert et la diffusion des connaissances et du savoir-faire.
- Contribuer à la valorisation des résultats de la recherche.
- Contribuer, notamment par la formation, à des programmes de coopération pour le développement.
- Effectuer des expertises scientifiques.
- Contribuer à la normalisation.

L'INRIA est implanté dans cinq régions :

- En Ile de France, une unité de recherche à Rocquencourt, ainsi que le siège de l'institut également localisé à Rocquencourt.
- En Bretagne, une unité de recherche à Rennes (créée en 1980).
- En Provence-Alpes-Côte d'Azur, une unité de recherche à Sophia-Antipolis (créée en 1982).
- En Lorraine, une unité de recherche à Nancy (créée en 1984).
- En Rhône-Alpes, une unité de recherche près de Grenoble (créée en 1992).

1.1.2 Présentation de l'INRIA de Sophia-Antipolis

Créée au coeur de la technopole Sophia-Antipolis en 1983, l'unité de recherche regroupe, sur ses sites de Sophia-Antipolis, Marseille et Montpellier, 500 personnes dont 380 scientifiques répartis au sein d'une trentaine d'équipes en partenariat avec le CNRS, plusieurs universités et grandes écoles. Leurs travaux portent sur la

conception et la programmation de systèmes informatiques performants, la représentation et la manipulation d'informations complexes, la création, la modélisation et la simulation d'expériences complexes. Ils permettent l'avancée des connaissances dans quatre grands domaines : 1)réseaux et systèmes, 2)génie logiciel et calcul symbolique,3)interaction homme-machine, 4)images, données, connaissances, simulation et optimisation des systèmes complexes.

1.1.3 Présentation de l'équipe ORION

ORION est une équipe pluri-disciplinaire, à la frontière des domaines de la vision par ordinateur, des systèmes à base de connaissances et du génie logiciel. L'objectif d'ORION est de concevoir et de développer des techniques et des logiciels pour d'une part, l'interprétation automatique d'images et d'autre part, la réutilisation et le pilotage automatique de programmes.

Axes de recherche Cet objectif est poursuivi en développant des langages d'expression des connaissances ainsi que des mécanismes d'apprentissage et de traitement de ces connaissances, adaptés à des classes de problèmes spécifiques. ORION se focalise sur l'étude des connaissances qui interviennent dans les deux types de problèmes étudiés :

- Connaissances sur les objets et les scénarios à reconnaître pour l'interprétation automatique d'images.
- Connaissances sur les programmes et leur utilisation pour le pilotage automatique de programmes.

Mon stage s'inscrit dans le premier problème étudié : connaissances sur les objets et les scénarios à reconnaître pour l'interprétation automatique d'images.

ORION étudie plus particulièrement les techniques de représentation des connaissances hybrides (à base de frames, de réseaux sémantiques et de règles de production) ainsi que les techniques de planification. Pour le raffinement de bases de connaissances, elle étudie des techniques d'apprentissage symbolique.

Les applications qui intéressent ORION relèvent principalement du domaine de la vision par ordinateur. A terme, ils souhaitent élargir le champ d'application des méthodes et techniques développées par ORION à d'autres domaines (traitement du signal, calcul scientifique).

1.2 Descriptif du sujet de stage

Ce stage se situe dans le cadre des travaux du projet ORION sur l'interprétation automatique de séquences vidéos pour la reconnaissance de comportements humains. L'équipe ORION a conçu une plate-forme d'interprétation vidéo VSIP (Visual Surveillance Intelligent Platform) permettant la détection d'objets mobiles, le suivi de leur trajectoire et la reconnaissance de certains comportements prédéfinis par les experts humains dans une base de scénarios. L'objectif de ce stage est de créer un outil qui permet de générer automatiquement des séquences vidéos à partir de modèles de scénario définis par les experts humains du domaine d'application. Ceci permettra aux experts de valider leur modélisation de scénarios en visualisant des séquences et aussi d'évaluer la plate-forme d'interprétation.

1.3 Objectifs et motivations du stage

Depuis plusieurs années, il y a un grand intérêt pour l'interprétation automatique de séquences vidéos. Plusieurs travaux se concentrent sur l'étude et l'identification de comportements humains [1], [2]. Notre but est le développement d'une plate-forme capable d'identifier des comportements humains prédéfinis par des experts du domaine d'application (par exemple, un directeur de la sécurité dans les banques)[3]. Par expérience nous savons qu'il y a plusieurs difficultés pour les experts de fournir une définition précise d'un modèle de scénario immédiatement. Par exemple, ils peuvent oublier une information importante ou ils ne sont pas capables d'exprimer clairement leurs besoins. La majeure partie du temps, c'est un processus itératif, c'est-à-dire qu'ils ne peuvent raffiner leurs modèles qu'une fois qu'ils ont vu une démonstration du système d'interprétation. La génération automatique d'animations 3D peut aider à améliorer ce procédé d'acquisition de connaissances. Les experts pourront visualiser par les animations 3D le résultat de leur modélisation de scénarios. Ils valideront leurs modèles, ajouteront ou supprimeront les données qui semblent nécessaires ou inutiles. La génération automatique d'animations 3D est également un moyen d'évaluer les algorithmes de vision. Des séquences synthétiques peuvent être produites à partir de ces animations 3D afin d'examiner et d'évaluer la fiabilité des algorithmes de vision. Beaucoup de vidéos peuvent être créées afin d'accentuer sous plusieurs configurations un problème non résolu d'un algorithme de traitement vision donné et afin d'évaluer la robustesse de cet algorithme sous ces diverses configurations.

Je suggère ainsi deux étapes pour produire des animations 3D à partir de modèles de scénarios comme représenté sur la figure 1. La première étape transforme un modèle de scénario (informations fournies par des experts comme des actions, des états, des contraintes, l'environnement) en un modèle de visualisation requis pour produire des animations 3D. Je propose dans ce rapport une méthode per-

mettant cette transition que j'appelle "transformation de scénario". Une deuxième étape produit des instances de scénarios à partir d'un modèle de visualisation : j'initialise les différents paramètres requis pour produire des animations 3D. Cette deuxième étape s'appelle "instanciation" et elle est responsable de l'initialisation des paramètres de la génération de trajectoires et des paramètres de rendu.

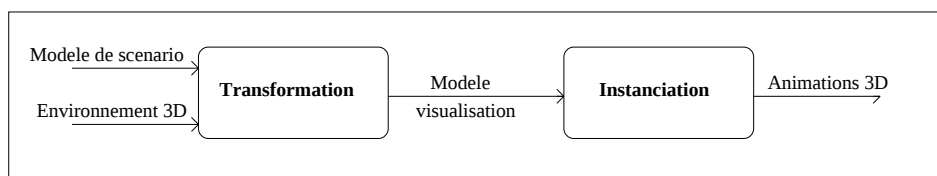


FIG. 1. Deux grandes étapes de génération d'animation 3D à partir de modèles de scénarios.

2 La plate-forme VSIP

2.1 Présentation

La plate-forme VSIP (Visual Surveillance Intelligent Platform) est un système pour l'interprétation automatique de séquences d'images. Son rôle est d'analyser automatiquement une scène donnée à partir de séquences d'images.

Elle est développée par l'équipe ORION. Elle prend en entrée un flux vidéo provenant d'une ou plusieurs caméras et fournit en sortie un message selon le comportement prédéfini qui a été reconnu. Le traitement des images s'effectue en deux étapes comme présenté sur la figure 2. Une fois les images de la caméra sont acquises à une cadence fixe et elles sont numérisées elles sont envoyées vers un logiciel capable de les traiter afin d'en extraire des caractéristiques numériques (module de traitement des séquences d'images). Ensuite, ces caractéristiques sont traitées dans l'optique d'obtenir une description sémantique et une reconnaissance de certains comportements (module de reconnaissance de scénario).

Le module de traitement de séquences peut lui même être décomposé en trois étapes (voir figure 3).

- Une étape de détection qui extrait de l'image courante les régions mobiles en mouvement, témoignant de l'existence possible d'objets mobiles.

- Une étape de classification des régions mobiles en différentes catégories d'objets mobiles tels que personnes, voitures...
- Une étape de suivi de ces objets mobiles un suivi d'une image à l'autre ou un suivi long terme (sur un grand nombre d'images) afin de calculer leurs propriétés temporelles (vitesse, trajectoire par exemple) et de les reconnaître.

Les motivations d'ordre applicatif sont multiples et concernent différents domaines. En surveillance d'activités humaines, le système d'interprétation a pour objectif de surveiller des zones dans les métros, les parkings, les banques, les supermarchés,... Le but est de détecter les comportements anormaux d'individus évoluant dans ces zones et de prévenir les situations dangereuses. Chleq et Thonnat [4] ont réalisé un système de ce type qui déclenche une alarme lorsqu'une situation à risque est détectée. En surveillance routière, l'objectif est de prévenir les situations dangereuses sur un réseau routier en détectant les incidents et en déclenchant des alarmes. Nagel [5] et ses collaborateurs ont développé Epex, un système dédié à l'interprétation de scènes routières à partir de séquences d'images .

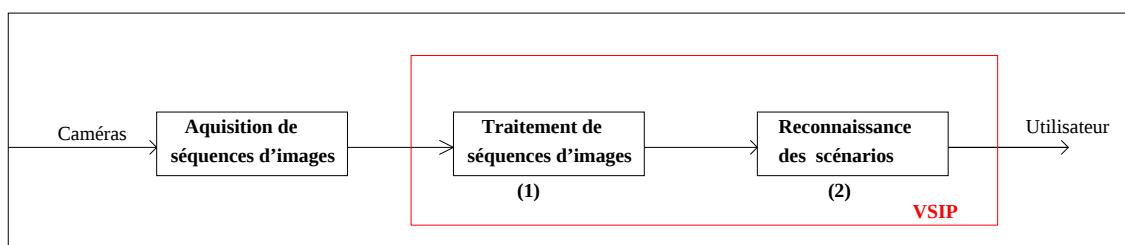


FIG. 2. Les deux principaux modules de la plate-forme VSIP.

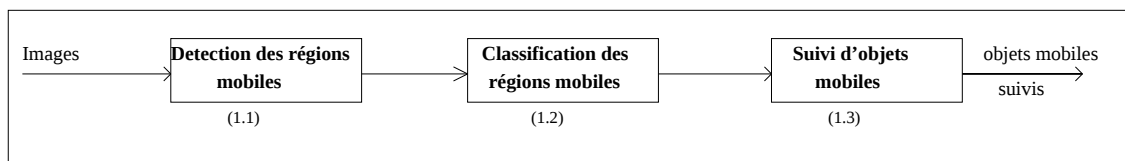


FIG. 3. Décomposition du module traitement de séquences d'images de la plate-forme VSIP.

2.2 Extension

L'objectif de mon stage est la réalisation d'un système qui permette la génération automatique d'animations 3D représentatives d'un modèle de scénario défini par l'expert qui servira à la fois pour l'évaluation des algorithmes d'interprétation de la plate-forme et pour la validation des modèles définis par l'expert. La figure 4 montre l'extension proposée de la plate-forme VSIP pour réaliser cet objectif.

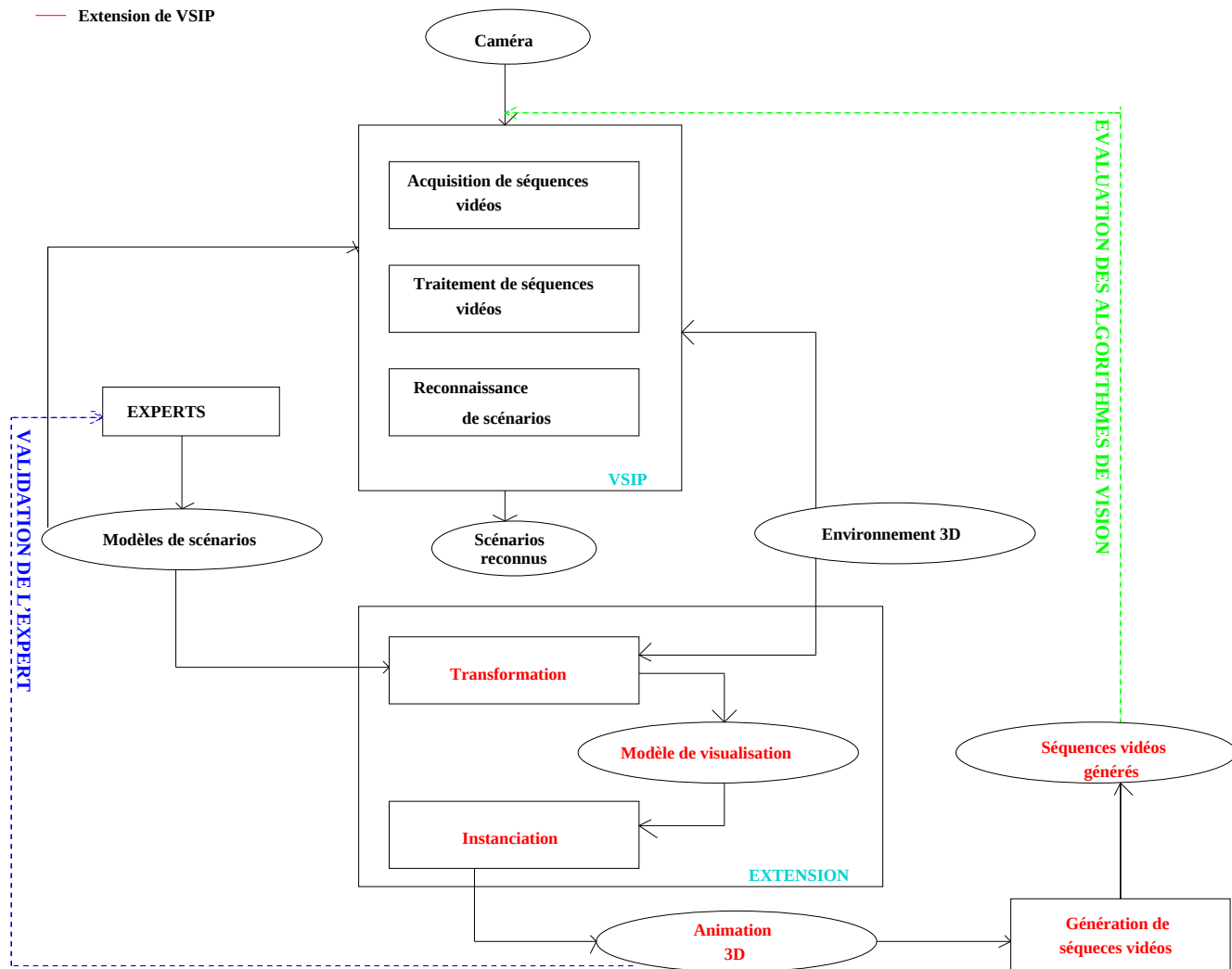


FIG. 4. Extension de la plate-forme VSIP

3 Génération d'animations 3D

3.1 Etat de l'art

Depuis plusieurs années, le problème de visualisation 3D a été étudié. Il y a eu beaucoup de travaux sur la visualisation d'une scène 3D à partir de sa description. Par exemple :

- A la faculté d'informatique de l'université de Toronto [6], les chercheurs produisent des animations 3D où des poissons et un nageur évoluent dans la mer. Ils ont en particulier modélisé différents comportements de poissons et leurs interactions dans un groupe.
- Craig.W.Reynolds [7] a présenté diverses solutions pour obtenir des personnages autonomes dans des animations de jeux. Il a décrit plusieurs comportements de direction tels que la poursuite, l'évasion et l'évitement d'obstacles.
- Au laboratoire d'informatique de l'institut Suisse de technologie de Lausanne [8], [9] des chercheurs ont modélisé des individus dans un musée, dans une rue et dans un supermarché. Ils ont également modélisé la réaction de personnes dans une situation de feu.
- Thalmann et al[10] ont simulé des animations 3D de personnes évoluant dans des couloirs et autour d'obstacles.
- L'équipe Siames de l'unité de recherche INRIA Rennes (Irisa), France [11] travaillent sur la synthèse d'images, l'animation, la modélisation et la simulation. Ils génèrent par exemple des animations dans lesquelles des véhicules et des individus évoluent (projet "virtual urban") ou des truites évoluent (projet "dynamic locomotion").
- Plusieurs travaux dans l'animation 3D liés à la robotique ont été également effectués. Nous pouvons citer [12] qui explique la méthode de champ potentiel utilisée pour la génération de trajectoires.
- L'équipe Robotvis de l'unité de recherche INRIA Sophia-Antipolis, France [13], [14], [15] fait de la recherche sur la synthèse d'images à partir de traitement d'images sur des séquences vidéos. Leurs objectifs est de visualiser le suivi d'un individu en train de courir.
- Le Robotics Institute, Carnegie Mellon University [16] à partir de caméras disposées sur tout le site de l'université, ont généré une animation 3D visualisant un groupe d'individus rentrant et sortant du site.

Tous ces travaux ont obtenu beaucoup de résultats dans le domaine des animations 3D d'une scène à partir de sa description.

Mon travail s'inspire de celui de Craig.W.Reynolds

L'originalité de mon travail est :

- L'introduction d'un modèle de visualisation qui joue le rôle de pont entre les scénarios définis par les experts humains et les animations 3D. Le contenu de ce modèle est étroitement lié aux besoins de l'expert et aux paramètres requis pour évaluer les algorithmes de vision.
- L'intérêt de ces animations 3D est double : validation du modèle défini par l'expert et génération de séquences pour l'évaluation de la plate-forme d'interprétation.
- L'entrée du système : un modèle de scénario défini par l'expert.

3.2 Contexte de la scène

3.2.1 Présentation du contexte

Le modèle du décor de la scène ou contexte de la scène comprend l'ensemble des informations contextuelles relatives à l'environnement 3D de la scène [17].

Le modèle de décor d'une scène qu'utilise l'équipe ORION contient notamment les informations géométriques 3D ainsi que les informations sémantiques de la scène.

Les informations géométriques Les informations géométriques concernent :

- Les zones : une décomposition de l'espace en zones. Nous avons décomposé par exemple l'espace de l'agence bancaire en 6 zones, définissant ainsi des zones particulières telle que la zone devant le coffre ou derrière le comptoir.
- La description 3D des objets du décor tels que table, chaise,...cette description renseigne sur les dimensions de l'objet, l'emplacement de sa base 2D dans la scène ainsi que sa hauteur.
- Les plans verticaux : un plan vertical est défini par ses deux points extrêmes (de début et de fin) et par sa hauteur. Un plan vertical est généralement un mur.

Les informations sémantiques Les informations sémantiques renseignent sur la nature de la zone par exemple zone d'entrée/sortie, zone à risque, sur la nature de l'objet. Un objet peut être transparent, opaque... ainsi que sur la nature des plans verticaux, plan vertical vitré ou pas ...les informations sémantiques renseignent aussi sur la couleur de ces différents éléments.

Toutes ces données sont définies dans un fichier XML comme présentés sur les figures 5, 6 et 7.

```
</Zone>
<Zone ident = "3" name = "CabinePilotage_RGB_137_137_133" place_name = "second_level" >
  <Ref_distance_list>
  </Ref_distance_list>
  <Properties_list>
  </Properties_list>
  <Outline_list>
    <Point x="686" y="-145" z="66" />
    <Point x="102" y="-145" z="66" />
    <Point x="102" y="145" z="66" />
    <Point x="686" y="145" z="66" />
  </Outline_list>
</Zone>
```

FIG. 5. Exemple de définition d'une zone de la scène.

```
</Equipment_obj>
<Equipment_obj ident = "3" name = "dossierSiegeLateralGauche_RGB_99_150_204" plane_name = "first_level" heigh = "86" >
  <Properties_list>
  </Properties_list>
  <Outline_list>
    <Point x="590" y="123" z="22" />
    <Point x="630" y="123" z="22" />
    <Point x="630" y="101" z="22" />
    <Point x="590" y="101" z="22" />
  </Outline_list>
</Equipment_obj>
```

FIG. 6. Exemple de définition d'un objet de la scène.

```
</Wall>
<Wall ident = "2" name = "murGaucheG_RGB_175_179_165" place_name = "ground" height = "97">
  <Properties_list>
  </Properties_list>
  <Start_point x="380" y="145" z="216" />
  <End_point x="509" y="145" z="216" />
</Wall>
```

FIG. 7. Exemple de définition d'un plan vertical de la scène.

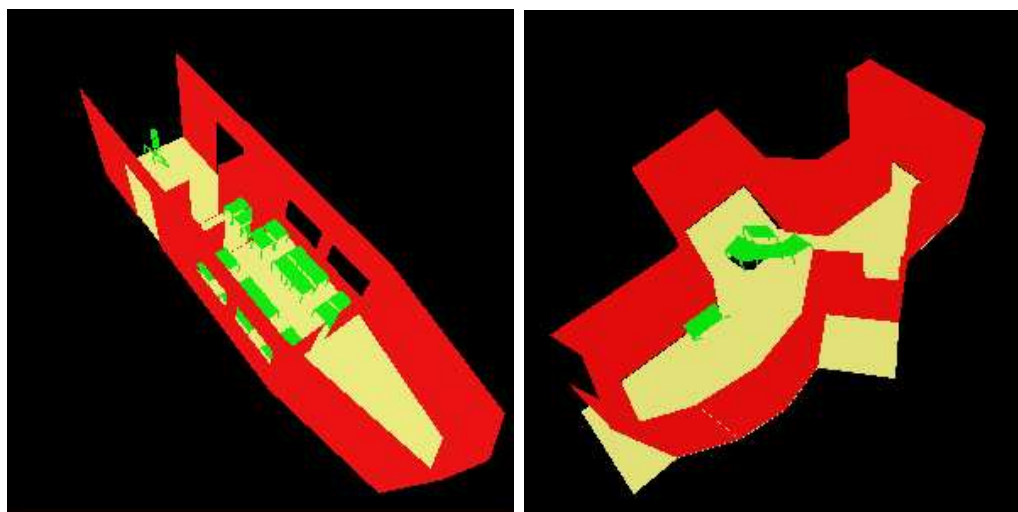


FIG. 8. Visualisation automatique du train et de la banque à partir du fichier contexte.

3.2.2 Visualisation du contexte

Pour une visualisation automatique du contexte de la scène, j'ai utilisé la librairie de visualisation 3D OpenSG [18]. Cette librairie est une sur-couche de OpenGL (elle se base sur les mêmes classes qu'OpenGL mais elle est de plus haut niveau). Pour l'implémentation, j'ai défini trois classes, chaque classe correspondant à un composant du contexte de la scène : une classe zone, une classe objet et une classe plan vertical.

La figure 8 présente les images du décor de l'agence bancaire et du train générées automatiquement à partir du contexte.

3.3 Représentation de scénarios

Dans cette section, je décris les modèles de scénarios donnés par les experts humains. J'utilise un formalisme de représentation qui prend en compte le domaine d'application de l'expert et le langage de description des scénarios [19]. Le modèle de scénario utilisé pour la reconnaissance de scénarios utilise les notions d'états, d'événements et de scénarios.

- Un état (state) est une propriété spatio-temporelle définie pendant un instant ou un intervalle de temps et qui caractérise à un instant donné un individu

déecté par une caméra.

- Un événement (event) est un ou plusieurs changements d'états à deux instants successifs.
- Le scénario peut être primitif (un seul changement d'état) ou composé (combinaison d'événements et états).

Pour la reconnaissance, un scénario est un ensemble de contraintes portant sur des individus évoluant dans la scène, sur des objets du décor et sur des événements (et sous scénarios) précédemment reconnus. Les contraintes temporelles sont exprimées à l'aide de relations liant le déroulement des différents états ou événements.

3.3.1 Modèle générique des scénarios

J'utilise un modèle générique de scénario formé des trois éléments suivants :

- La liste des objets physiques (physical objects) : tous les objets réels présents dans la scène observée par les caméras. Un objet physique peut être un objet mobile (personne, voiture) ou un objet statique de l'environnement. Un objet statique est défini par une connaissance à priori de l'environnement et peut être une zone d'intérêt (devant la porte) ou un équipement (table, chaise).
- La liste des composants (components) constituant le scénario : liste des différents états, événements et sous scénarios composant le scénario. Chaque composant est formé de trois variables : (deux objets physiques) et (un état ou un événement ou un sous-scénario). Chaque composant doit être formé au moins d'un objet physique non statique (personne, voiture,...).
- Les contraintes : présente la relation entre les différents objets physiques et/ou les composants. Les contraintes nous renseignent souvent sur le déroulement chronologique dans le temps des composants.

Un exemple d'état, d'événement et de scénario sont présentés sur les figures 9, 10 et 12 respectivement.

primitive state inside_zone physical_objects: (p: person) (z: zone) constraints: (p in z)
--

FIG. 9. Exemple d'état.

primitive event move_close_to physical_objects: (p: person) (eq : equipment) components: (c1: primitive_state far_from (p, eq) (c2: primitive_state close_to (p, eq)) constraints: (c1 before c2) (c1 duration <=2)
--

FIG. 10. Exemple d'événement.

3.4 Transformation de scénarios

Dans cette section je décris la méthode utilisée pour transformer les modèles de scénarios définis par les experts humains en modèles de visualisation contenant les informations nécessaires pour générer des animations 3D. Le contenu de ce modèle de visualisation est étroitement lié aux besoins de l'expert et aux paramètres utiles pour évaluer les algorithmes de vision. Je commence par expliquer la méthode utilisée pour transformer le modèle de scénario en modèle de visualisation, j'illustrerai par la suite la méthode par deux exemples, un exemple de transformation d'un scénario simple et d'un scénario plus compliqué.

3.4.1 Méthode de transformation de scénarios

L'idée pour réaliser la transformation de scénario est de transformer le modèle de scénario défini par l'expert en un graphe orienté. Les noeuds du graphe correspondent aux composants (components). Un composant peut être un état, un événement ou un sous scénario. Chaque composant est transformé en deux noeuds. Un noeud correspond à la date de commencement du composant, l'autre correspond à la date de fin. Ces deux noeuds sont reliés par un arc orienté qui indique la durée du composant, la valeur de cet arc orienté est initialisée à $]0, \infty[$ vu que je suppose initialement que chaque composant possède une durée dans le temps qui va de 0 à l'infini.

Une fois tous les noeuds du graphe sont formés l'étude des contraintes imposées par l'expert permet de relier entre eux deux noeuds correspondant à deux composants différents avec la bonne valeur de l'arc qui les relie. L'étude des contraintes permet également de réduire la dimension de l'intervalle défini par l'arc reliant deux

noeuds du même composant.

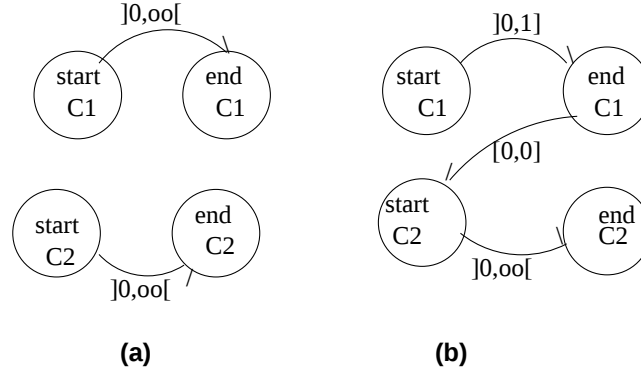


FIG. 11. Exemple de présentation graphique : avant et après l'étude des contraintes

Dans le modèle de scénario défini par l'expert humain, les contraintes sont définies par les opérateurs suivants : before, during, start, finish, meet, overlap, before_meet (before ou meet). Par exemple, la contrainte (c1 meet c2) et (c1 duration ≤ 1) induit les transformations présentées sur la figure 11.

Une fois le graphe formé avec les valeurs adéquates des arcs, je transforme chaque arc en une équation linéaire. En se référant par exemple à la figure 11 la traduction de l'arc reliant start c1 et end c1 est

$$end_{c1} = start_{c1} + a \quad (1)$$

avec a appartenant à l'intervalle $]0,1[$.

Le parcours de tout le graphe modélisant le modèle de scénario me permet d'obtenir un système d'équations linéaires relié aux instants.

Pour déterminer les positions clés, l'idée est également de transformer le modèle de scénario défini par l'expert en un système d'équations linéaires. Ce système d'équations est lui par contre relié aux composants non pas aux contraintes comme c'était le cas pour le système précédent relatif aux instants.

Dans le modèle de scénario défini par l'expert, les composants sont présentés par les opérateurs suivants : far from, close to, inside_zone, outside_zone.

Si on note P1 le premier objet physique du composant, P2 le deuxième objet physique du même composant et pos_P1/pos_P2 leurs positions respectives, j'obtiens les équations linéaires suivantes :

- far_from (P1, P2) est équivalent à
pos_P1 = pos_P2 + dist ; dist > distMin .
- close_to (P1, P2) est équivalent à
pos_P1 = pos_P2 + dist ; dist < distMax.
- inside_zone (P1, P2=zone) est équivalent à
pos_Actor1 = centre_zone + dist ; dist < distMin.
- outside_zone (P1, P2= zone) est équivalent à
pos_Actor1 = frontiere_zone + dist ; dist > distMax.

Je définis donc notre modèle de scénario comme suit :

- Un système d'équations linéaires relatif aux instants.
- Un système d'équations linéaires relatif aux positions.
- Trajectoires.
- Le contexte de la scène : informations géométriques de l'environnement 3D observé par une ou plusieurs caméras.
- Les paramètres de rendu tel que le contraste, l'illumination, la position et le nombre d'éclairage.
- La vitesse des acteurs de la scène.

Le contenu de ce modèle est étroitement lié aux besoins de l'expert et aux paramètres requis pour évaluer les algorithmes de vision.

3.4.2 Exemple de scénario simple

Je considère le scénario simple d'une personne qui s'approche d'un équipement (voir figure 10)

Le système relatif aux contraintes :

$$S1 = \begin{cases} tf_s_close = ti_s_close + A; A \in]0 \ 2[\\ ti_s_close = tf_s_far + B; B > 0 \end{cases}$$

Le système relatif aux composants :

$$S2 = \begin{cases} pos_p_s_far = pos_eq + distMax. \\ pos_p_s_close = pos_eq + distMin. \end{cases}$$

pos_eq est donnée par une connaissance à priori de l'environnement 3D de la scène observée.

3.4.3 Exemple de scénario complexe

Dans cet exemple, l'expert a modélisé un acte de vandalisme. Une personne approche un équipement, puis s'éloigne et se rapproche à nouveau. Ce va-et-vient se produit dans le cas où le vandale est perturbé par la présence d'une personne inconnue pendant son acte de vandalisme (voir figure 12).

scenario vandalism

physical_objects:
 (p: person) (eq : equipment)

components:
 (e1: primitive_event move_close_to (p, eq))
 (e2: primitive_event stay_at (p, eq))
 (e3: primitive_event move_away_from (p, eq))
 (e4: primitive_event move_close_to (p, eq))
 (e5: primitive_event stay_at (p, eq))

constraints:
 (e1 before_meet e2) (e2 before_meet e3)
 (e3 before_meet e4) (e4 before_meet e5)

FIG. 12. Scénario de vandalisme.

Le graphe correspondant est présenté sur la figure 13

Le système d'équations linéaires relatif aux contraintes

$$S = \begin{cases} ti_st_at1 = tf_mv_close1 + A ; A \geq 0 \\ ti_mv_away = tf_st_at1 + B ; B \geq 0. \\ ti_mv_close2 = tf_mv_away + C ; C \geq 0. \\ ti_st_at2 = tf_mv_close2 + D ; D \geq 0. \end{cases}$$

Le système d'équations linéaires relatif aux instants

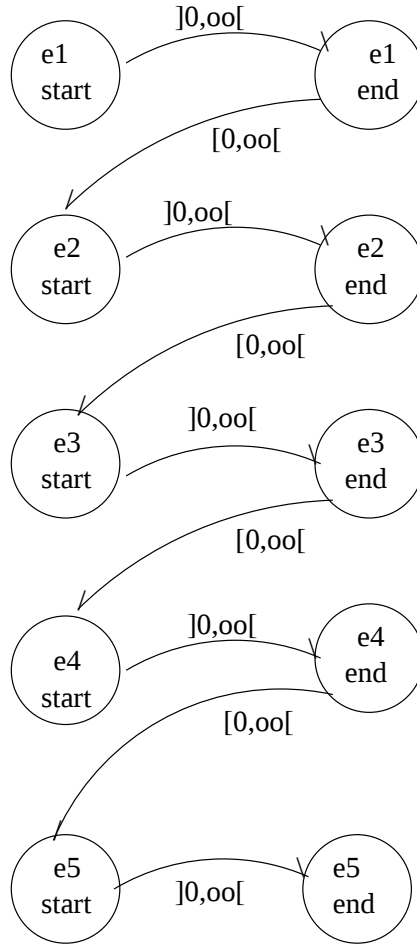


FIG. 13. Graphe du scénario vandalisme.

$$S = \begin{cases} \text{pi_mv_close1} = \text{position_eq} + \text{distMax.} \\ \text{pf_mv_close1} = \text{pos_eq} + \text{distMin.} \\ \text{p_st_at1} = \text{pos_eq} + \text{distMin.} \\ \text{pi_mv_away_from} = \text{pos_eq} + \text{distMin.} \\ \text{pf_mv_away_from} = \text{pos_eq} + \text{distMax.} \\ \text{p_st_at2} = \text{pos_eq} + \text{distMin.} \end{cases}$$

3.5 Instanciation

Il s'agit dans cette partie de spécifier comment je génère des instances de scénarios à partir du modèle de visualisation que j'ai défini. Il s'agit de choisir les

positions et instants clés, de générer une trajectoire convenable et de spécifier les différents paramètres de rendu.

La variabilité dont je dispose pour choisir tous ces paramètres me permet de générer beaucoup d'animations 3D pour le même modèle de scénario.

3.5.1 Positions et instants clés

Les systèmes d'équations linéaires relatif aux positions et aux instants ont plusieurs solutions. Chaque animation correspond à l'une des solutions.

Je choisis pour toute équation de notre système, une valeur de l'élément variable dans l'intervalle qui lui correspond. Je résous ensuite le système par une méthode classique telle que la méthode du Pivot de Gauss. (une explication de cette méthode est donnée en annexe)

Exemple Un exemple de transformation des systèmes S1 et S2 définis dans la section 3.4.2 est :

$$S11 = \begin{cases} tf_s_close = ti_s_close + 1 \\ ti_s_close = tf_s_far + 10 \end{cases}$$

$$S22 = \begin{cases} pos_p_s_far = pos_eq + 20. \\ pos_p_s_close = pos_eq + 2. \end{cases}$$

3.5.2 Génération de trajectoires

Principe Dans le paragraphe précédent, j'ai décrit comment extraire, à partir d'un modèle de visualisation, les données numériques relatives aux instants et aux positions clés. Dans cette section, je vais d'écrire la méthode utilisée pour générer des trajectoires ayant ces données comme entrée.

La méthode utilisée est basée sur la méthode du champ potentiel. D'une part les obstacles (chaise, table...) appliquent une force répulsive sur la personne. D'autre part, le point d'arrivée ou le but applique une force attractive qui tend vers zéro quand la personne approche du but.

Pour les murs du contexte, afin d'éviter les problèmes de minimum local et vu que dans certains scénarios on cherche à ce que la personne longe le mur, l'idée est que dès qu'une personne approche un mur elle change de direction en formant un angle

de 90 degrés avec sa direction précédente comme représenté sur la figure 14. De plus, dans les cas où il y a plus d'une personne, afin d'éviter le problème de collision entre deux personnes chacun dispose d'un champ potentiel répulsif autour de lui d'une intensité relativement faible.

Un autre point est à préciser, lorsqu'une personne doit suivre une autre (cas d'un vol par exemple), que les deux personnes aient les mêmes positions de departs et d'arrivée n'implique pas nécessairement un suivi de personne. L'idée est donc, si je note P1 la personne à suivre et P2 la personne qui suit, P2 aura à chaque instant t des positions voisines de celles qu'avaient P1 à l'instant $t-a$ ($a>0$).

En conclusion, pour les mêmes positions et instants clés trouvés dans l'étape précédente, je peux générer plusieurs animations 3D en changeant la direction initiale des personnes de la scène ou en sélectionnant différentes valeurs des paramètres impliqués dans la génération de trajectoires tels que l'intensité de la force répulsive appliquée par les obstacles.

Implémentation de la méthode Pour l'implémentation de la méthode, j'ai travaillé sur la projection 2D de la scène. Chaque objet est présenté par un disque circonscrit à sa base.

Les personnes sont également assimilées à un disque. Le rayon de ce disque est à préciser en fonction de la taille choisie de la personne.

J'ai opté pour un disque pour la présentation des éléments de la scène vu qu'il est plus simple de définir un champ potentiel pour un disque que pour une forme plus compliquée. Les champs potentiels répulsifs et attractifs ont les expressions respectives :

$$U_{repul} = \begin{cases} 1/2.k_{repul}.\left(\frac{1}{r} - \frac{1}{r_o}\right) & \text{si } r < r_o \\ 0 & \text{si } r > r_o \end{cases}$$

$$U_{attra} = 1/2.k_{attra}.(c - c_{goal})^2 \quad (2)$$

$k_{repul} > 0, k_{attra} > 0$.

r est la distance à l'obstacle, r_o est la distance d'influence des obstacles.

c est la coordonnée du mobile, c_{goal} est la coordonnée du but

3.5.3 Les paramètres de rendu

Un autre degré de liberté pour générer des animations 3D correspondant au même modèle de visualisation consiste à modifier les paramètres de rendu tels que

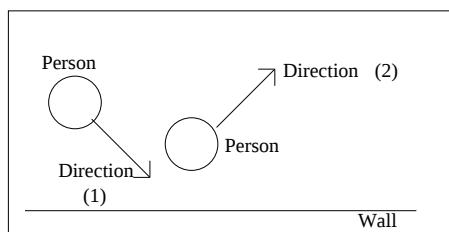


FIG. 14. Direction d'une personne de la scène s'approchant d'un mur

le contraste ou l'éclairage.

Les paramètres de rendu ont une grande influence sur l'obtention d'animations réalistes et par conséquent sur la génération de séquences vidéos complexes. Cette complexité et cette variabilité sont nécessaires pour évaluer la fiabilité et la robustesse des algorithmes de vision. Par exemple, une séquence vidéo synthétique composée de plusieurs personnes en mouvement ayant des couleurs totalement différentes ne présente pas de difficultés majeures pour les algorithmes de tracking. Par contre les algorithmes de tracking peuvent échouer dans le cas d'une séquence vidéo synthétique avec des personnes ayant la même couleur ou des couleurs qui se rapprochent.

Le contraste présente également une influence majeure sur la détection de personnes. Un faible contraste provoque souvent des problèmes de détection.

Les ombres et les éclairages sont également d'autres exemples qui influencent la détection.

Avoir de nombreuses séquences vidéos qui peuvent servir de données d'entrée pour évaluer les algorithmes de vision est d'une importance primordiale. D'abord, elles évitent le long processus d'annotation manuelle des vidéos afin d'obtenir des données de vérité de terrain. En second lieu, elles permettent à des experts de bien comprendre les problèmes courants des algorithmes de vision, de les diagnostiquer et de développer par conséquent de nouveaux algorithmes qui sont de plus en plus robustes.

4 Résultats et discussion

- Dans une agence bancaire, j'ai obtenu des animations 3D comme présentées sur la figure 16 en prenant en entrée le modèle de scénario défini par la figure 15. Il s'agit d'un scénario de vol dans une agence bancaire. Les deux animations 3D présentées sur la figure sont différentes vu que les trajectoires des deux personnes intervenant dans la scène diffèrent d'une animation à l'autre. Cette différence est obtenue par un changement de la direction initiale des

```
scenario
physical_objects:
((employee: Person), (robber: Person), (z1: Entrance),
 (z2: Back_Counter), (z3: Infront_Counter), (z4: Safe))
components:
((c1: primitive_state inside_zone(employee, z2))
 (c2: primitive_event changes_zone(robber, z1, z3))
 (c3: primitive_event changes_zone(employee, z2, z4)
 (c4: primitive_event changes_zone(robber, z3, z4)))
constraints:
((c2 during c1)
 (c2 before c3)
 (c1 before c3)
 (c2 before c4)
 (c4 during c3))
```

FIG. 15. Scénario de la banque

personnes de la scène. Pour les deux animations, j'ai gardé cependant les mêmes positions de départ et d'arrivée ainsi que les mêmes instants obtenus par résolution des systèmes d'équations linéaires définis dans notre modèle de visualisation.

Les paramètres de rendu restent également inchangés dans les deux animations.

- Pour le modèle de visualisation, un retour de l'expert pour un scénario de vol dans un train (voir figure 17) m'a conduit à ajouter un paramètre vitesse dans le modèle de visualisation. En effet l'expert, en voyant l'animation 3D générée à partir du modèle qu'il a défini, a jugé que la vitesse du voleur était trop faible comparée à la réalité.

D'autres résultats se trouvent à l'adresse web suivante :

<https://www-sop.inria.fr/orion/personnel/Jihene.Bannour/>

- Acceptation d'un article intitulé "Generation of 3D Animations from Scenario Models" pour la conférence VIIP (une copie de l'article est donnée en annexe).

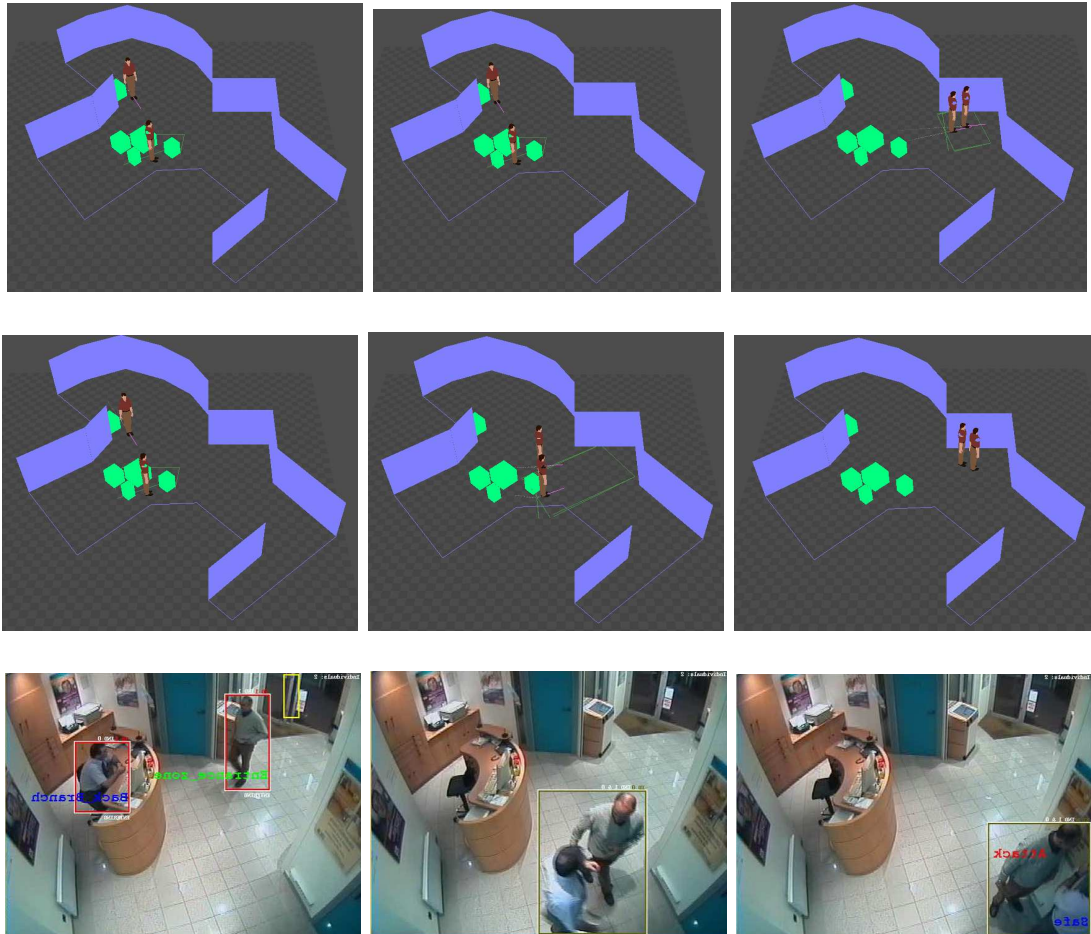


FIG. 16. Deux animations 3D pour le même modèle de visualisation et la vidéo correspondante

5 Conclusion et perspectives

5.1 Conclusion

Dans ce rapport, j'ai présenté la méthode utilisée pour générer des animations 3D à partir de modèles de scénarios définis par les experts humains. J'ai insisté sur les deux grandes étapes nécessaires pour résoudre ce problème : transformation de scénario et instanciation. Grâce à cette méthode les experts seront capables de valider les scénarios qu'ils ont défini. Ce résultat est très important vu qu'il permet de diminuer le temps d'acquisition des connaissances et d'affiner leur contenu.

5.2 Perspectives

Les travaux futurs se concentreront sur les problèmes de rendu et sur le réalisme des animations 3D. Ceci permettra de mieux évaluer les algorithmes de vision. Par exemple, on prendra en considération la posture des personnes présentes dans la scène (lever la main, s'asseoir, marcher, courir...), les problèmes d'ombres, d'éclairage,...

On essaiera aussi de tester les méthodes développées sur des scénarios autre que ceux de l'agence bancaire et du train.

On améliorera également notre modèle de visualisation en fonction des retours des experts.

Références

- [1] V.Vu, F.Brémont, and M.Thonnat, "Video interpretation : human behaviour representation and on-line recognition," in *The Sixth International Conference on Knowledge-Based Intelligent Information and Engineering Systems*, (sophia antipolis, France), 2002.
- [2] M.Arens and H.-H.Nagel, "Behavioral knowledge representation for the understanding and creation of video sequences," in *Proceedings of the 26th German Conference on Artificial Intelligence*, (Hamburg, Germany), September 2003.
- [3] B.Georis, F.Brémont, M.Maziere, and M.Thonnat, "A video interpretation platform applied to bank agency monitoring," in *Intelligent Distributed Surveillance Systems Workshop*, (London, UK), February 23 2004.
- [4] Chleq and Thonnat, "Realtime image sequence interpretation for video-surveillance applications," in *International conference on Image Processing (ICIP'96). Processing IEEE ICIP'96*, (Lausanne, Switzeland), pp. Vol 2. pp 801–804, 1996.
- [5] Nagel, "The representation of situations and their recognition from image sequences," 1998.

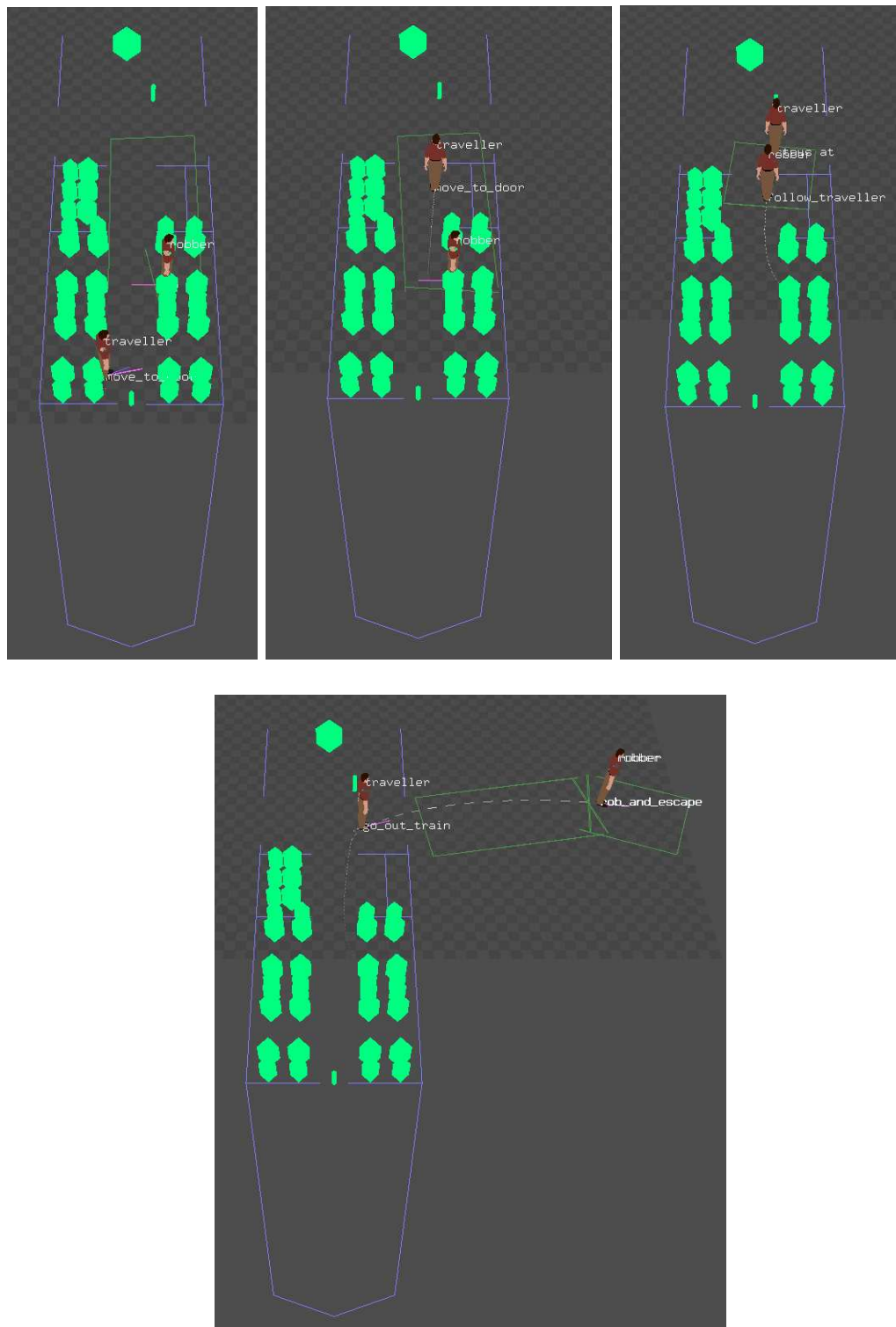


FIG. 17. Animation d'un vol dans un train

- [6] D. Terzopoulos, "Artificial life for computer graphics," in *Communication of the ACM*, 42(8), pp. 32–42, 1999.
- [7] C. W.Reynolds, "Steering behaviours for autonomous characters," in *Sony Computer Entertainment America*, (Foster City, California), June 1 1999.
- [8] L.Bezault, R.Boulic, and N.M.Thalmann, "An interactive tool for the design of human free walking trajectories," in *Computer animation*, 1992.
- [9] R.Boulic, D.Thalmann, and N.M.Thalmann, "A global human walking model with real-time kinematic personification," in *Visual Computer*, 1990.
- [10] Thalmann, Daniel, Olivier, and M.-T. Nadia, "A vision-based approach to behavioral animation," in *Journal of Visualization and Computer Animation* (J. of Visualisation and C. Animation, eds.), pp. 18–21, 1990.
- [11] "[http ://www.inria.fr/recherche/equipes/siames.fr.html](http://www.inria.fr/recherche/equipes/siames.fr.html),"
- [12] H.Hadded, M.Khatib, S.Lacroix, and R.Chatila, "Reactive navigation in outdoor environments using potentiel field," (Toulouse , France), 1998.
- [13] Q. Delamarre and O. Faugeras, "3d articulated models and multiview tracking with silhouettes," in *ICCV 99*, (INRIA, Sophia-Antipolis), 1999.
- [14] Q. Delamarre and O. Faugeras, "Suivi multi-caméras de personnes et modèles 3d articulés," in *RFIA 00*, (INRIA, Sophia-Antipolis), 2000.
- [15] "[http ://www-sop.inria.fr/robotvis/robotvis-fra.html](http://www-sop.inria.fr/robotvis/robotvis-fra.html),"
- [16] R. T.Collins and T. Kanada, "Multi-camera and visualization for surveillance and sports," in *The Fourth International Workshop on Cooperative Distributed Vision*, (Robotics Institute Carnegie Mellon University), 2001.
- [17] F.Brémont and M.Thonnat, "Issues in representing context illustrated by scene interpretation applications," in *In proc. of the Int'l and Interdisciplinary Conf. on Modeling and Using Context (CONTEXT-97)*, 1997.
- [18] "www.opensg.org,"
- [19] V.Vu, F.Bremont, and M.Thonnat, "Temporal constraints for video interpretation," in *The 15-th European Conference on Artificial Intelligence*, (sophia antipolis, France), 2003.

6 Annexe

6.1 Méthode du Pivot de Gauss implémentée

Considérons le système $n \times n$ de n d'équations linéaires à n inconnues $x_1, x_2, \dots, x_n$:

$$S1 = \begin{cases} a_{1,1}x_1 + a_{1,2}x_2 + \dots + a_{1,n}x_n = b_1 \\ a_{2,1}x_1 + a_{2,2}x_2 + \dots + a_{2,n}x_n = b_2 \\ \dots \\ \dots \\ a_{n,1}x_1 + a_{n,2}x_2 + \dots + a_{n,n}x_n = b_n \end{cases}$$

où les $a_{i,j}$ et les b_i sont des réels donnés, i et j variant de 1 à n . Le système peut s'écrire $AX = B$ où A désigne la matrice des $a_{i,j}$, X la matrice colonne des x_i et B la matrice colonne des b_i . On suppose que le système est de Cramer : il admet une unique solution. C'est à dire que le déterminant de A est non nul.

La méthode de Gauss consiste à trianguler la matrice A correspondant au système $S1$. Le système $S1$ aura la forme suivante :

$$S2 = \begin{cases} a_{1,1}x_1 + a_{1,2}x_2 + \dots + a_{1,n}x_n = b_1 \\ a_{2,2}x_2 + a_{2,3}x_3 + \dots + a_{2,n}x_n = b_2 \\ a_{3,3}x_3 + \dots + a_{3,n}x_n = b_3 \\ \dots\dots\dots \\ a_{n-1,n-1}x_{n-1} + a_{n,n}x_n = b_{n-1} \\ a_{n,n}x_n = b_n \end{cases}$$

La matrice A est alors devenue :

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & 0 \\ a_{22} & a_{23} & \dots & 0 \\ O & a_{33} & \dots & \\ \vdots & \vdots & \ddots & 0 \\ O & 0 & \dots & \end{pmatrix}$$

On a fait apparaître des zéros au-dessous des termes diagonaux de $i = 2$ à $i = n - 1$.

Dans ces conditions, on obtient tout d'abord $x_n = b_n/a_n$, puis $x_{n-1}, \dots, x_2, x_1$.

Description de l'algorithme Quitte à permuter les lignes, on peut supposer que a_{11} est non nul. Si cela s'avère impossible, l'inconnue x_1 est arbitraire et le système n'est pas de Cramer. Pour éliminer x_1 en ligne 2, il suffit de lui retirer $a_{21}/a_{11} \times L_1 : L_2$ donne $L_2 - a_{21}/a_{11} \times L_1$.

Le terme constant b_2 subit la même transformation : b_2 donne $b_2 - a_{21}/a_{11} \times b_1$. D'une façon générale, pour éliminer x_1 dans les lignes 2 à n , donc faire apparaître des zéros au-dessous de a_{11} , on procède à la transformation :

L_i donne $L_i - a_{i1}/a_{11} \times L_1$ et b_i donne $b_i - a_{i1}/a_{11} \times b_1$.

On cherche à éliminer maintenant x_2 dans les lignes 3 à n : cela revient à réitérer le procédé sur la matrice extraite de A

$$A' = \begin{pmatrix} a_{22} & a_{23} & \dots \\ 0 & a_{33} & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

et sur le membre de droite (matrice colonne des constantes).

On procède à la transformation L_i donne $L_i - a_{i2}/a_{22} \times L_2$, i variant de 3 à n .

Si a_{22} s'était avéré nul, on aurait permuté les lignes jusqu'à trouver un a_{k2} non nul. Si cela n'est pas possible, alors x_2 est arbitraire : le système ne serait pas de Cramer.

Le procédé se poursuit pour éliminer x_k dans les lignes k , pour $k < n$. Les éléments diagonaux servant de diviseur sont appelés pivots. Remarquer que lors de l'élimination de x_k , les lignes de rang inférieur ne sont pas modifiées.

6.2 Article accepté

Article accepté pour la conférence IASTED-VIIP (Visualization, Imaging, and Image Processing) en Espagne.