Carsim : un système pour convertir des textes en scènes tridimensionnelles animées

Carsim: A System to Convert Texts into Animated 3D Scenes

Pierre Nugues[1] Sylvain Dupuy[2] Arjan Egges[3]

[1] LTH, Institut d'informatique, Université de Lund, Box 118, S-221 00 Lund, Suède
[2] ISMRA, 6, boulevard du Maréchal Juin, F-14050 Caen
[3] MIRALab, Université de Genève, 24, rue du Général Dufour, CH-1211 Genève, Suisse

Pierre.Nugues@cs.lth.se Sylvain.Dupuy@free.fr egges@miralab.unige.ch

Résumé

Cet article décrit un système de conversion de textes en scènes tridimensionnelles animées. Les textes sont des constats d'accidents de voitures rédigés par un des conducteurs. Le processus de conversion comporte de deux étapes. Un premier module d'extraction d'informations crée une description tabulaire de l'accident et un second module de visualisation produit une scène 3D et l'anime.
Nous décrivons l'architecture du système et la structure des fiches qui vont contenir l'information extraite. Nous décrivons ensuite le module d'extraction d'informations qui se charge de la détection des objets statiques et des véhicules, de leurs directions initiales, de la suite de mouvements et des collisions. Nous présentons de façon succincte le module de visualisation. Nous concluons par des copies d'écran d'animations et les résultats que nous avons obtenus.

Mots Clef

Conversion de textes en scenes, extraction d'informations, discours, espace et mouvement.

Abstract

This paper describes a system to convert written descriptions of car accident reports into animated 3D scenes. Descriptions are short narratives written by one of the drivers after the accident. The text-to-scene conversion process consists of two stages. An information extraction system creates a tabular description of the accident and a visual simulator generates and animates the scene.
We first outline the structure of the text-to-scene conversion and the template structure. We then describe the information extraction system: the detection of the static objects and the vehicles, their initial directions, the chain of events and the collisions. We succinctly present the visualization module. We show snapshots of the car animation output and we conclude with the results we obtained.

Keywords

Text-to-scene conversion, information extraction, discourse, space and motion.

1 La conversion de textes en scènes tridimensionnelles

Les images facilitent la compréhension d'une situation et la communication d'une idée [20], [8], [27]. Nous l'avons tous remarqué, il est souvent plus facile d'expliquer des phénomènes physiques, des théorèmes mathématiques ou de démêler des structures de toutes sortes en utilisant un schéma ou un graphique que des mots.
La conversion d'un texte en une scène consiste à recréer les idées visuelles et les images mentales qu'il contient sous la forme d'une composition géométrique bi ou tridimensionnelle. Cette conversion doit combiner des constructions à la fois visuelles et linguistiques. Michel Denis [9] remarque à cet effet que

> *L'image et le langage constituent deux modes de représentations [...] fortement différenciées, mais dont la coopération est requise dans de nombreuses formes de fonctionnement cognitif.*

Les images tridimensionnelles animées sont probablement supérieures aux graphiques statiques 2D pour visualiser des informations. Elles peuvent reproduire une scène concrète plus exactement et rendent possible l'exécution d'une suite d'actions. Quand les images tridimensionnelles sont couplées à des environnements virtuels, elles permettent à des utilisateurs de naviguer dans le monde 3D et d'interagir avec les objets qu'il contient.
Nous ne prétendons pas ici qu'une image puisse remplacer un texte. Cependant, il est probable que la production d'une scène visuelle à partir d'un texte aide à comprendre l'information qu'il contient, la rende plus concrète et la complète.

L'intérêt des techniques automatiques est de faire ces conversions plus rapidement et plus facilement.

2 Travaux antérieurs

La conversion automatique d'un texte en une scène a fait l'objet d'un certain nombre d'études. NALIG [1], [21] est une première réalisation dont l'objectif était de recréer des scènes bidimensionnelles. L'un des buts était de modéliser les relations entre l'espace et les prépositions dans des expressions en italien du type sujet, préposition, objet. D'après ce qui est décrit dans les articles, NALIG ne traitait pas de phrases complètes et encore moins de textes.
WordEye [7] est un autre exemple à la fois récent et impressionnant qui génère des scènes tridimensionnelles animées à partir de textes courts. WordsEye dispose de 12.000 objets graphiques ce qui donne une idée de son ambition. Il utilise des ressources linguistiques telles que l'analyseur de dépendances de Collins [6] et le réseau lexical WordNet [14]. L'interprétation du récit est fondée sur une extension des grammaires de cas et des inférences sur la structure de l'environnement [24]. WordsEye, cependant, ne traite pas des histoires réelles. Les récits ressemblent plutôt à des contes de fées, apparemment inventés par les auteurs.
CogViSys [22] est un dernier exemple dont l'idée d'origine était de synthétiser textes à partir d'une séquence d'images vidéos. Les auteurs ont ensuite jugé qu'il pouvait être intéressant de faire le chemin inverse et de générer des séquences vidéos à partir de textes. Le moteur logique du convertisseur de textes [2] est fondé sur la *Théorie de représentation du discours* de Kamp et Reyle (DRT) [19]. CogViSys est limité à la visualisation de manœuvres simples à un carrefour telles que celle décrite par ces deux phrases : *A car came from Kriegstrasse. It turned left at the intersection* [3]. Les auteurs ne donnent pas plus de détails sur leur corpus, ni de précisions sur leurs résultats.
Enfin, TRAINS [15], Ulysse [5], [17] et Persona [4] sont des projets voisins qui comportent une interaction écrite ou orale dans des mondes virtuels.

3 Le système CarSim

CarSim [26], [10] est un prototype de visualisation et d'animation des scènes tridimensionnelles à partir de descriptions écrites. Il opère dans le domaine des accidents de la route et nous l'avons appliqué à un corpus de 87 constats fournis par la compagnie d'assurance MAIF.
Les constats sont « amiables ». Il n'y a normalement pas de blessés et les deux conducteurs se sont mis d'accord sur les conditions de l'accident. Les descriptions prennent la forme de courts récits rédigés par un des conducteurs après le sinistre. Elles correspondent à des accidents relativement simples bien qu'un petit nombre d'entre elles soient difficiles voire impossibles à interpréter. L'objectif de CarSim est de « rejouer » ces accidents symboliquement dans un monde virtuel.
L'architecture du système comporte deux parties qui communiquent par une représentation formelle de l'accident. La structure de cette représentation est une fiche semblable à celles d'autres systèmes d'extraction d'informations ([25], [18], entre autres). Nous avons conçu cette fiche de telle sorte que les données qu'elle renferme permettent de reproduire visuellement et d'animer les entités d'un accident.
La première partie de CarSim est un module linguistique qui extrait l'information à partir du constat et remplit les éléments d'information de la fiche. La deuxième partie est un visualiseur qui prend comme entrée la fiche remplie, crée les entités visuelles et les anime (Figure 1).

4 Deux exemples de constats

Les deux textes qui suivent sont des exemples de constats :

> *Véhicule B venant de ma gauche, je me trouve dans le carrefour, à faible vitesse environ 40 km/h, quand le véhicule B, percute mon véhicule, et me refuse la priorité à droite. Le premier choc atteint mon aile arrière gauche, sous le choc, et à cause de la chaussée glissante, mon véhicule dérape, et percute la protection métallique d'un arbre, d'où un second choc frontal.*
>
> Texte A4.

> *Je roulais sur la partie droite de la chaussée quand un véhicule arrivant en face dans le virage a été complètement déporté. Serrant à droite au maximum, je n'ai pu éviter la voiture qui arrivait à grande vitesse.*
>
> Texte A8.

Le texte A4 décrit deux collisions : d'abord entre deux voitures et puis entre une des voitures et un arbre. C'est une suite de chocs relativement complexe pour ce corpus mais le récit est explicite.
Les descriptions ne sont pas toujours aussi directes et « franches ». Beaucoup de rapports contiennent des litotes, des négations ou des descriptions visant à enjoliver ou à amoindrir la responsabilité du conducteur. Cela empêche souvent de les comprendre sans faire intervenir une connaissance approfondie des conditions de conduite ou sans essayer de deviner l'état d'esprit du conducteur. Le rapport A8 ne mentionne pas de collision par exemple. Quelques rares constats sont même complètement incompréhensibles à la seule lecture du texte.

5 La représentation des connaissances

Comme les autres systèmes d'extraction d'informations, CarSim repose sur des fiches définies au préalable pour représenter le texte. Ceci signifie que le contenu est réduit à une structure tabulaire, suffisante pour décrire ce qui s'est produit et pour permettre une conversion en une scène icônique (symbolique).

FIG. 1 – *L'architecture de CarSim.*

Chaque fiche se compose de trois listes [11] :

- une liste d'objets statiques identifie le type de la route et les objets qui jouent un rôle dans la collision (arbres, feux de circulation, etc.) ;
- une liste d'objets dynamiques décrit les véhicules, voitures ou camions, impliqués dans l'accident, et enfin
- une liste ordonnée de collisions décrit la suite d'impacts entre les objets dynamiques et éventuellement statiques.

Le Tableau 1 contient la fiche correspondant au constat A4. La liste STATIC contient les deux objets statiques importants : la route et l'arbre. La forme de la route est choisie parmi quatre cas possibles : ligne droite, tournant à gauche ou à droite et carrefour – straight_road, left_turning_curve, right_turning_curve, crossroads.
La liste DYNAMIC décrit deux véhicules : l'énonciateur – narrator – et l'autre véhicule. Chaque VÉHICULE – ou acteur – détecté dans le texte est représenté par un ensemble d'attributs : le nom du véhicule, sa direction initiale et une série d'actions atomiques décomposant son déplacement.
Les constats d'assurance identifient habituellement les véhicules par A et B. Quand cette information n'est pas disponible, la ligne est remplie avec « énonciateur ». L'autre véhicule est identifié comme le B.
La fiche décrit les positions relatives des véhicules au début du récit. En effet, la majeure partie des textes ne donne pas de coordonnées géographiques exactes. Les directions initiales sont contenues dans la ligne INITDIRECTION. Elles sont choisies parmi les points cardinaux : ouest, est, nord et sud.
La suite d'événements (EVENT) correspond aux mouvements des véhicules. Les actions possibles actuellement sont avancer, tourner à gauche, tourner à droite, s'arrêtez, changer de file vers la droite, vers la gauche, et dépasser (forward, turn_left, turn_right, change_lane_right, change_lane_left, overtake, stop).
Enfin, la liste ACCIDENT donne la suite des collisions.
Le Tableau 1 décrit une première collision entre l'énonciateur et le véhicule B, puis entre l'énonciateur et l'arbre. Les deux parties en cause sont appelées ACTEUR et VICTIME bien que ce ne soit pas une interprétation des responsabilités. Quand c'est possible, la fiche contient aussi les parties touchées. COORD est l'endroit plausible de la collision où (0, 0) désigne le centre de la scène, ici le centre du carrefour.

Nous avons évalué les possibilités de description de ce formalisme pour représenter les accidents du corpus. Nous avons extrait les informations et rempli les fiches manuellement et nous avons constaté qu'environ 60% des accidents pouvaient être reproduits. Les 40% restant correspondent à des configurations de routes ou à des scénarios d'accidents plus complexes.

6 L'extraction des informations

Le module d'extraction d'informations (EI) remplit les éléments de la fiche que nous avons décrite précédemment. Le déroulement du traitement consiste à analyser le texte en utilisant un analyseur syntaxique partiel, à déterminer parmi des situations prototypiques les conditions de l'accident et à affiner les résultats par des modules sémantiques. Le module d'EI utilise le contenu littéral de certaines expressions qu'il trouve dans le texte ou bien infère l'environnement et les actions.

6.1 L'extraction des objets statiques

La première étape de l'extraction d'informations est une analyse syntaxique partielle. Elle utilise des règles syntagmatiques écrites dans la notation *Definite Clause Grammar* du langage Prolog. L'analyse détecte les groupes nominaux, les groupes verbaux et les groupes prépositionnels ainsi que quelques expressions figées. Les marques ou les modèles des véhicules sont des exemples de ces expressions ainsi que les groupes de mots *véhicule A* ou *véhicule B*.
Une deuxième étape extrait les objets statiques. Elle détecte la configuration de la route, les panneaux routiers, les feux et les obstacles. Elle utilise un lexique du domaine. Pour détecter un carrefour, le module recherche des mots tels que *croisement*, *carrefour* ou *intersection*; pour les virages à droite ou à gauche, il recherche des mots comme *tournant*, *virage*, *courbe*, etc.
La détermination de la configuration correcte de la route est un élément très important du processus. Il conditionne en grande partie le résultat final.

6.2 La détection des acteurs

Les acteurs sont les véhicules ou les personnes jouant un rôle dans l'accident. Dans la plupart des constats, la description de l'accident est présentée du point de vue de l'énonciateur. Celui-ci décrit ce qui s'est produit en utilisant la première personne. Dans quelques autres rapports, la description est impersonnelle et emploie des phrases telles que *le véhicule A a heurté le véhicule B*.

Constat A4

```
STATIC [
  ROAD [
    KIND = crossroads;
  ]
  TREE [
    ID = tree;
    COORD = (-4, 4);
  ]
]
DYNAMIC [
  VEHICLE [
    ID = v\’{e}hicule_b;
    KIND = car;
    INITDIRECTION = south;
    CHAIN [
      EVENT [
KIND = driving_forward;
      ]
    ]
  ]
  VEHICLE [
    ID = narrator;
    KIND = car;
    INITDIRECTION = east;
    CHAIN [
      EVENT [
KIND = driving_forward;
      ]
    ]
  ]
]
ACCIDENT [
  COLLISION [
    ACTOR = v\’{e}hicule_b, unknown;
    VICTIM = narrator, left_side;
    COORD = (-1.5, 1.5);
  ]
  COLLISION [
    ACTOR = narrator, left_side;
    VICTIM = tree, unknown;
    COORD = (-1.5, 1.5);
  ]
]
```

Constat A8

```
STATIC [
  ROAD [
    KIND = turn_left;
  ]
]
DYNAMIC [
  VEHICLE [
    ID = narrator;
    KIND = car;
    INITDIRECTION = east;
    CHAIN [
      EVENT [
KIND = driving_forward;
      ]
    ]
  ]
  VEHICLE [
    ID = v\’{e}hicule;
    KIND = car;
    INITDIRECTION = south;
    CHAIN [
      EVENT [
KIND = driving_forward;
      ]
    ]
  ]
]
ACCIDENT [
  COLLISION [
    ACTOR = narrator, unknown;
    VICTIM = v\’{e}hicule, unknown;
    COORD = (-1.5, 1.5);
  ]
]
```

TAB. 1 – *Les fiches correspondant aux constats A4 et A8.*

Dans CarSim, nous associons les acteurs d'un accident aux groupes nominaux. La détection des acteurs revient alors essentiellement à une résolution de co-références.
Par défaut, un premier acteur appelé l'énonciateur est créé. Puis, le module de détection des acteurs crée autant d'acteurs qu'il trouve de noms différents dans les groupes nominaux. Ces noms doivent être sémantiquement compatibles avec des véhicules ou des conducteurs : les noms propres, les identifiants tels que le véhicule A ou véhicule B, les marques de véhicules, les modèles comme *Fiat Punto*, *Renault Clio*, des noms de personne, etc.
En cas d'échec, le détecteur d'acteurs recherche les objets statiques qui peuvent être impliqués dans une collision. Dans ce dernier cas, un seul acteur est créé. Enfin, si aucun des deux cas précédents n'est vérifié, le détecteur d'acteur crée un deuxième véhicule de défaut.
La résolution des co-références associe tous les pronoms et les adjectifs possessifs à la première personne à l'énonciateur. Elle associe les pronoms et les adjectifs possessifs à la troisième personne aux autres acteurs en utilisant un critère de récence. Si aucun pronom à la première personne n'est trouvé, l'énonciateur est supprimé.
Une des suppositions de l'algorithme est que le nombre probable d'acteurs est le plus souvent deux et parfois un ou trois. Il permet à cet algorithme très simple d'atteindre une détection de 80% des acteurs dans les textes du corpus.

6.3 Les directions initiales

Les directions initiales sont détectées grâce à une petite ontologie de domaine de la route et des mouvements de voitures [16]. Quand ils commencent à se déplacer, les véhicules peuvent être sur un même axe ou avoir des directions orthogonales.
Comme en général, il n'y a pas de référence absolue, par convention, l'énonciateur vient de l'est dans notre représentation. Par conséquent, les voitures venant de la droite viendront du nord dans l'orientation symbolique. Les voitures venant de la gauche seront situées au sud. Les voitures en face viendront de l'ouest. Les voitures derrière ou se déplaçant avant la même direction viendront également de l'est.
Quelques constats indiquent des endroits géographiques et des directions comme

Me rendant à Beaumont-sur-Oise... (Texte A1)

Quand c'est possible, on extrait les noms de villes ou de directions tels que *Beaumont-sur-Oise* pour remplir l'élément STARTSIGN dans la liste VÉHICULE de la fiche. Ces noms sont visualisés comme des panneaux de direction en forme de flèche.
Un système plus élaboré pourrait recaler les voitures sur un meilleur modèle de routes en utilisant les noms des rues et éventuellement de vraies cartes. Ceci nécessiterait un traitement complémentaire de toutes les informations du constat. Ici, nous nous sommes limités à ne traiter que le texte libre.

Pour détecter les directions initiales, l'algorithme recherche des couples de groupes verbaux et de directions. Les directions sont indiquées en général par des compléments circonstanciels de lieu. Les verbes correspondent à *venir de*, *surgir de*, *arriver sur*, etc. Les compléments circonstanciels correspondent à *sur la droite*, *en sens inverse*, etc. Le sujet de la phrase est alors unifié avec un des acteurs détectés précédemment. Cela permet d'initialiser les déplacements des véhicules.

6.4 La détection des collisions

Le module de détection des collisions utilise un lexique de verbes de chocs. Ces verbes correspondent à ceux nous avons observés dans les constats : *percuter*, *heurter*, *casser*, *accrocher*, etc. ainsi qu'à quelques autres que nous avons ajoutés. Nous les avons classés selon leur sens et s'ils entraînent un deuxième choc comme *pousser* ou *projeter*. Le module considère également des négations telle que celle de la phrase *Je n'ai pas pu éviter*...
Le module extrait le sujet et le complément d'objet direct de chaque verbe de choc. Il vérifie si le verbe est au passif ou à l'actif et il détecte les couples sujet-verbe, verbe-objet, verbe-agent et verbe-cible. Le couple verbe-cible correspond à certaines constructions verbe-préposition comme *cogner sur quelque chose* où le groupe prépositionnel pourrait être considéré comme un objet dans d'autres langues, par exemple *hit something* en anglais.
Le module de détection des collisions identifie les relations grammaticales en utilisant leurs propriétés topologiques à la manière de FASTUS ([18], p. 397). Les principes de l'algorithme sont les suivants :

- Le sujet d'un verbe actif est le dernier groupe nominal avant le verbe, à condition qu'il corresponde à un acteur.
- Le complément d'objet est le premier groupe nominal après le verbe si c'est un acteur ou un objet statique.
- L'agent est un groupe prépositionnel où le nom est un acteur valide et la préposition est *par*.
- La " cible " est un groupe prépositionnel où le nom est un acteur valide et avec une construction spécifique verbe-préposition.

Dans le cas d'une forme passive, l'agent remplace le sujet. On détecte les sujets et les compléments d'objet grâce à des règles DCG. Le sujet du verbe est associé à l'une des classes d'acteurs. S'il existe un complément d'objet et qu'il corresponde à une classe d'acteurs différente, alors l'objet est rattaché à cette classe d'acteur.
S'il y a une contradiction, par exemple quand le sujet et l'objet correspondent à la même classe ou que l'objet n'apparaît dans aucune classe d'acteurs, l'algorithme tente de la résoudre en employant des règles d'inférence de telle sorte que l'objet soit rattaché à une autre classe. Cette méthode

est acceptable car dans la plupart des constats, il n'y a que deux acteurs.
Le système crée autant de collisions qu'il y a des verbes de chocs à moins que deux collisions ne fassent participer exactement les mêmes acteurs. Le système enlève alors les doublons. En effet, dans certains contats, le conducteur se répète et mentionne deux fois la même collision.

6.5 La détection des chaînes de mouvements

Chacun des véhicules réalise une ou plusieurs actions différentes avant la collision. Ces actions sont codées comme une liste d'événements (EVENT) contenus dans l'élément CHAIN des objets dynamiques. Les actions que le visualiseur peut réaliser sont avancer, tourner à gauche ou à droite, changer de file à gauche ou à droite, dépasser et s'arrêter.
Dans les constats, ces actions correspondent à des verbes de mouvement tels que *démarrer*, *rouler*, *circuler*, *avancer*, *tourner*, *virer*, etc. Le module de détection d'événements extrait ces verbes du texte et il crée un nouvel événement pour chaque occurrence d'un des verbes. Il l'affecte à la chaîne de mouvements du sujet du verbe à condition qu'il corresponde à un acteur valide.
Certains verbes décrivent un enchaînement de plusieurs mouvements comme *redémarrer* qui indique que le conducteur s'est arrêté avant. L'événement est alors une suite de deux actions : stop et driving_forward.
Le mécanisme d'extraction ajoute les événements à la chaîne dans l'ordre où ils sont exposés dans le texte. Ceci ne correspond pas toujours à la vraie chronologie. Cependant pour une grande majorité des accidents, il n'y a pas de « retour en arrière » et l'ordre réel des événements correspond à leur ordre dans le texte.

7 La synthèse de scènes

Le module de visualisation a comme entrée les fiches remplies. À partir d'une fiche, il synthétise une scène tridimensionnelle symbolique et anime les véhicules [11], [13], [12].
L'algorithme de génération de scène place les objets statiques dans l'espace et calcule les mouvements des véhicules. Il utilise des règles d'inférence pour vérifier la cohérence de la fiche et pour estimer les positions de début et de fin de mouvement.
Le visualiseur utilise un planifieur pour créer les trajectoires. La première étape de la planification calcule les positions de début et de fin des mouvements en fonction des directions initiales, de la configuration de la route et de la suite d'actions comme s'il n'y avait pas eu d'accident. Un deuxième module modifie ces trajectoires pour insérer les collisions en fonction de la partie ACCIDENT de la fiche.
Cette décomposition en deux étapes se justifie par les descriptions qu'on observe dans la plupart des constats. Les conducteurs commencent généralement le récit de l'accident comme si c'était un déplacement normal qui ensuite est bouleversé par des conditions anormales. Enfin, le module temporel du planifieur affecte des intervalles de temps aux segments de la trajectoire.

8 Résultats et perspectives

Nous avons évalué le système CarSim sur tous les constats du corpus de la MAIF. La méthode que nous avons adoptée est plus simple que la métrique classique des systèmes d'extraction d'informations qui utilise la précision et le rappel. Nous avons jugé la chaîne complète de traitement en comparant le rendu visuel que CarSim produisait d'un texte à la compréhension qu'une personne (l'un des auteurs de cet article) pouvait avoir de l'accident et les images mentales qu'il pouvait en former.
Les Figures 3 et 4 montrent des images des scènes synthétisées pour les constats A4 et A8. Comme on peut le voir, l'évaluation comporte une part de subjectivité.
En utilisant les algorithmes d'extraction d'informations et de visualisation que nous avons décrits ici, nous avons pu obtenir une simulation plausible ou acceptable de 22 à 28 constats. Ceci représente de 25 à 32% du corpus. Ces chiffres doivent être comparés à la capacité de représentation de notre modèle de fiche et à celle de notre système de visualisation. Elles fixent à 60%, la limite supérieure du nombre de textes que nous pouvons simuler.
Il est à noter aussi que nous avons réalisé cette évaluation sur notre corpus de développement. L'utilisation d'un corpus de test distinct diminuerait très probablement ces résultats.
Les causes des échecs sont les suivantes :

- Les descriptions spatiales sont parfois imprécises. Certains constats sont difficiles à interpréter même pour des hommes ou des femmes. L'auteur s'appuie probablement fortement sur le schéma qu'il joint au rapport. CarSim ne se sert pas de ce schéma et ne peut alors pas positionner correctement les véhicules.

- La base géométrique d'objets statiques est limitée à quelques objets routiers. Elle ne contient pas de représentation de ronds points, trottoirs, accès d'autoroute, barrières, etc.

- Certaines actions de véhicule ne sont pas modélisées comme la marche arrière.

Pour finir, nous vérifions la validité de notre architecture avec d'autres types de résumés d'accidents. Nous avons appliqué des parties de l'algorithme d'extraction d'informations et le visualiseur à des descriptions en anglais. Nous avons utilisé un petit corpus des résumés d'accidents de la route disponible au site Internet du Conseil national pour la sécurité des transports aux États-Unis (National transportation safety board) [23]. Nous développons actuellement un nouveau système pour le suédois avec un corpus de textes beaucoup plus important.
Bien que nos résultats actuels soient loin d'être parfaits, ils démontrent la pertinence de notre démarche. Une stratégie fondée sur l'extraction d'informations permet de convertir des textes réels dans des scènes tridimensionnelles. Les

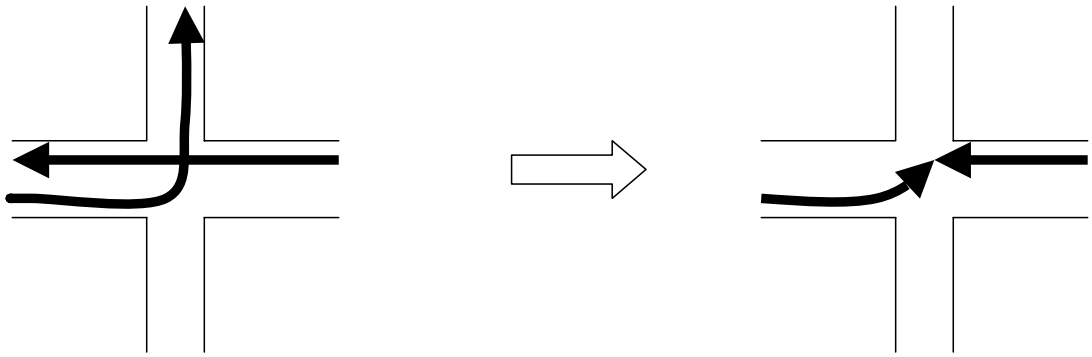

FIG. 2 – *La planification des trajectoires.*

résultats obtenus montrent que l'architecture de CarSim peut s'appliquer au français et à d'autres langues.
À notre connaissance, CarSim est le seul système de conversion de textes en scènes fonctionnant avec des récits non-inventés.

Remerciements

Le projet Carsim est partiellement financé par le programme Språkteknologi de Vinnova (`www.vinnova.se`), contrat 2002-02380.

Références

[1] Giovanni Adorni, Mauro Di Manzo, and Fausto Giunchiglia. Natural language driven image generation. In *Proceedings of COLING 84*, pages 495–500, Stanford, California, 1984.

[2] Michael Arens and Artur Ottlik. Automatische Analyse natürlichsprachlicher Texte zur Generierung von synthetischen Bildfolgen. Technical report, Universität Karlsruhe (TH), May 2000.

[3] Michael Arens, Artur Ottlik, and Hans-Hellmut Nagel. Natural language texts for a cognitive vision system. In Frank van Harmelen, editor, *ECAI2002, Proceedings of the 15th European Conference on Artificial Intelligence*, Lyon, July 21-26 2002.

[4] Gene Ball, Dan Ling, David Kurlander, John Miller, David Pugh, Tim Skelly, Andy Stankosky, David Thiel, Maarten van Dantzich, and Trace Wax. Lifelike computer characters: the Persona project at Microsoft. In Jeff M. Bradshaw, editor, *Software Agents*. AAAI/MIT Press, 1997.

[5] Olivier Bersot, Pierre-Olivier El Guedj, Christophe Godéreaux, and Pierre Nugues. A conversational agent to help navigation and collaboration in virtual worlds. *Virtual Reality*, 3(1):71–82, 1998.

[6] Michael Collins. A new statistical parser based on bigram lexical dependencies. In *Proceedings of the ACL 1996*, pages 184–191, 1996.

[7] Bob Coyne and Richard Sproat. Wordseye: An automatic text-to-scene conversion system. In *Proceedings of the Siggraph Conference*, Los Angeles, 2001.

[8] Michel Denis. Imagery and thinking. In Cesare Cornoldi and Mark A. McDaniel, editors, *Imagery and Cognition*, pages 103–132. Springer Verlag, 1991.

[9] Michel Denis. Rapport scientifique du groupe cognition humaine. Technical report, LIMSI, Orsay, 1996.

[10] Sylvain Dupuy, Arjan Egges, Vincent Legendre, and Pierre Nugues. Generating a 3D simulation of a car accident from a written description in natural language: The Carsim system. In *Proceedings of The Workshop on Temporal and Spatial Information Processing, 39th Annual Meeting of the Association of Computational Linguistics*, pages 1–8, Toulouse, July 7 2001. Association for Computational Linguistics.

[11] Arjan Egges. Generating a 3D simulation of a car accident from a formal description. Technical report, ISMRA Caen and Universitet Twente, 2000.

[12] Arjan Egges, Anton Nijholt, and Pierre Nugues. Carsim: Automatic 3D scene generation of a car accident description. Technical Report CTIT-01-08, The CTIT center of the University of Twente, Enschede, 2001.

[13] Arjan Egges, Anton Nijholt, and Pierre Nugues. Generating a 3D simulation of a car accident from a formal description. In Venetia Giagourta and Michael G. Strintzis, editors, *Proceedings of The International Conference on Augmented, Virtual Environments and Three-Dimensional Imaging (ICAV3D)*, pages 220–223, Mykonos, Greece, May 30-June 01 2001.

[14] Christiane Fellbaum, editor. *WordNet: An electronic lexical database*. MIT Press, 1998.

[15] George Ferguson, James F. Allen, and Bradford W. Miller. Trains-95: Towards a mixed-initiative planning assistant. In Brian Drabble, editor, *Proceedings of the Third International Conference on AI Planning Systems (AIPS-96)*, pages 70–77, 1996.

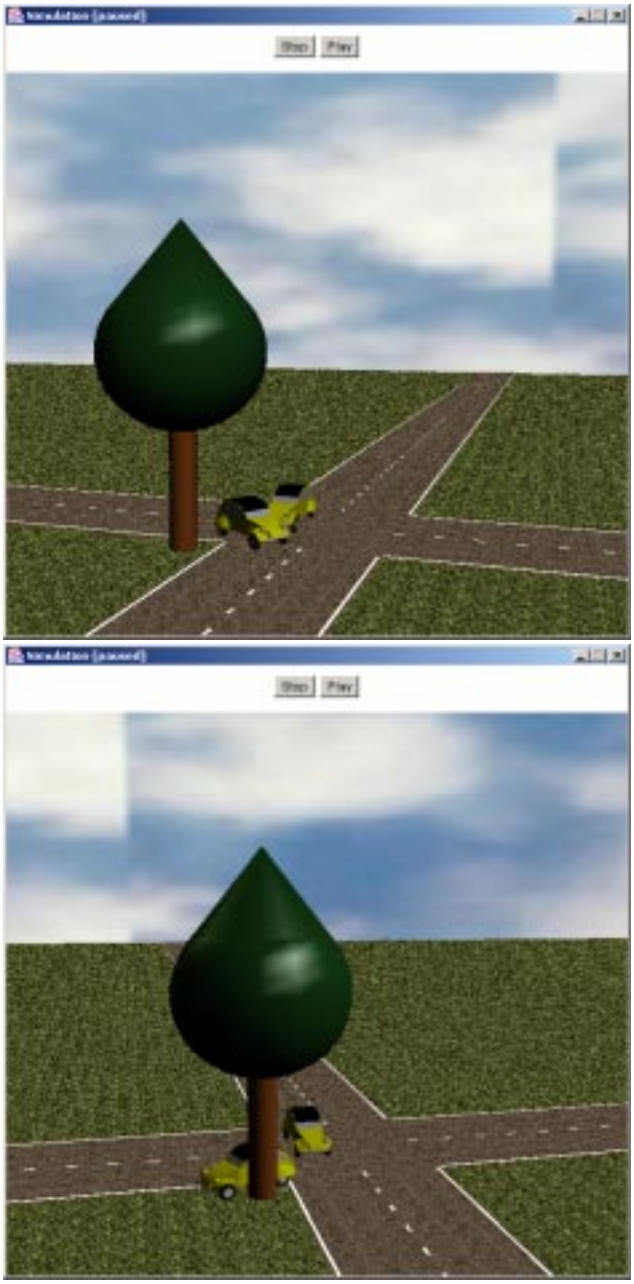

FIG. 3 – *Visualisation du texte A4 (Corpus MAIF).*

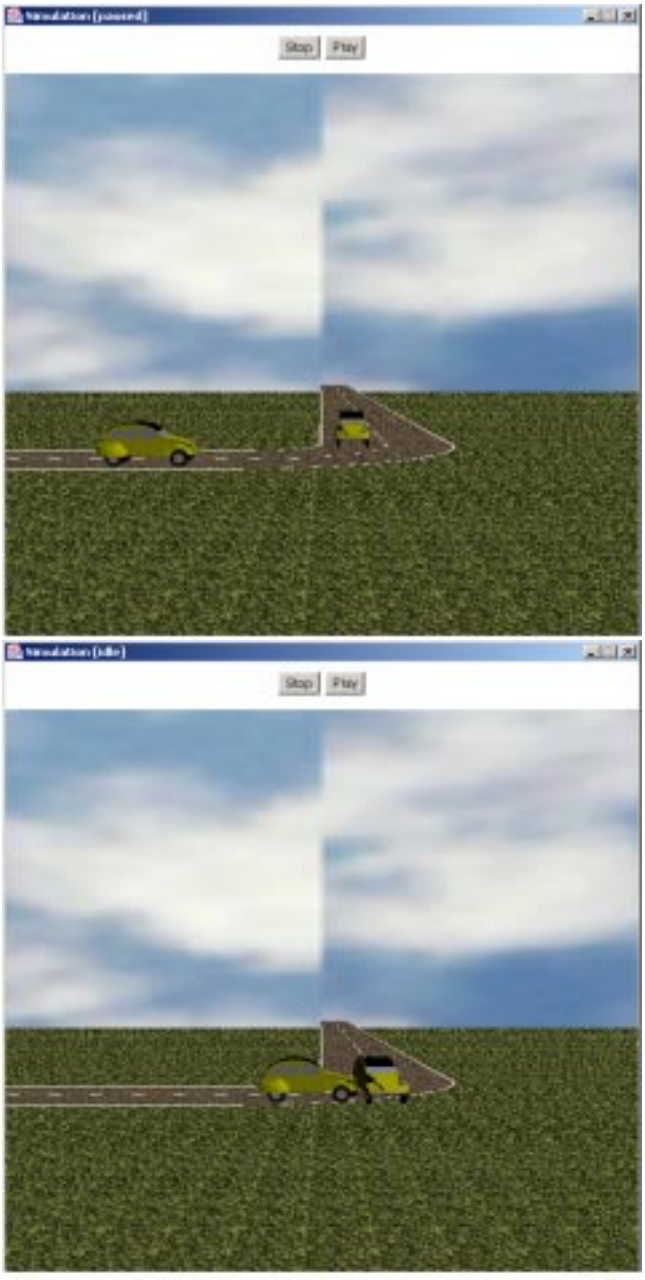

FIG. 4 – *Visualisation du texte A8 (Corpus MAIF).*

[16] Simon Le Gloannec, Pierre Aubeuf, and Cédric Métais. Extraction d'informations pour la simulation en 3D de textes de constats d'accidents. Technical report, ISMRA, Caen, 2001. Rapport de projet.

[17] Christophe Godéreaux, Pierre-Olivier El Guedj, Frédéric Revolta, and Pierre Nugues. Ulysse: An interactive, spoken dialogue interface to navigate in virtual worlds. Lexical, syntactic, and semantic issues. In John Vince and Ray Earnshaw, editors, *Virtual Worlds on the Internet*, chapter 4, pages 53–70, 308–312. IEEE Computer Society Press, Los Alamitos, California, 1999.

[18] Jerry R. Hobbs, Douglas Appelt, John Bear, David Israel, Megumi Kameyama, Mark Stickel, and Mabry Tyson. Fastus: A cascaded finite-state transducer for extracting information from natural-language text. In Emmanuel Roche and Yves Schabes, editors, *Finite-State Language Processing*, chapter 13, pages 383–406. MIT Press, 1997.

[19] Hans Kamp and Uwe Reyle. *From Discourse to Logic*. Kluwer, Dordrecht, 1993.

[20] Stephen Michael Kosslyn. *Ghosts in the Mind's Machine*. Norton, New York, 1983.

[21] Mauro Di Manzo, Giovanni Adorni, and Fausto Giunchiglia. Reasoning about scene descriptions. *IEEE Proceedings - Special Issue on Natural Language*, 74(7):1013–1025, 1986.

[22] Hans-Hellmut Nagel. Toward a cognitive vision system. Technical report, Universität Karlsruhe (TH), http://kogs.iaks.uni-karlsruhe.de/CogViSys, 2001.

[23] Ola Åkerberg, Hans Svensson, Bastian Schulz, and Pierre Nugues. Carsim: An automatic 3D text-to-scene conversion system applied to road accident reports. In *Proceedings of the Research Notes and Demonstrations of the 10th Conference of the European Chapter of the Association of Computational Linguistics*, pages 191–194, Budapest, Hungary, April 12-17 2003. Association for Computational Linguistics.

[24] Richard Sproat. Inferring the environment in a text-to-scene conversion system. In *Proceedings of the K-CAP'01*, October 22-23 2001.

[25] Beth Sundheim, editor. *MUC-3: Proceedings of the Third Message Understanding Conference*, San Francisco, 1991. Morgan Kaufmann.

[26] Fabrice Tabordet, Fabrice Pied, and Pierre Nugues. Scene visualization and animation from texts in a virtual environment. *CC AI. The journal for the integrated study of artificial intelligence, cognitive science and applied epistemology*, 15(4):339–349, 1999. Special issue on visualization. Guest editor: Tom G. De Paepe.

[27] Edward R. Tufte. *Visual Explanations: Images and Quantities, Evidence and Narrative*. Graphic Press, 1997.