A multimodal interpreter for 3D visualization and animation of verbal concepts

Coline Claude-Lachenaud[1,2], Éric Charton[1], Benoît Ozell[1], Michel Gagnon[1]

[1]École Polytechnique de Montréal
Montréal, H3T 1J4 , Québec (Canada)
{coline.claude-lachenaud, eric.charton, benoit.ozell, michel.gagnon}@polymtl.ca

[2]École des Mines de Nantes
Nantes, (France)
coline.claude-lachenaud@eleve.emn.fr

Abstract

We present an algorithm intended to visually represent the sense of verb related to an object described in a text sequence, as a movement in 3D space. We describe a specific semantic analyzer, based on a standard verbal ontology, dedicated to the interpretation of action verbs as spatial actions. Using this analyzer, our system build a generic 3D graphical path for verbal concepts allowing space representation, listed as SelfMotion concepts in the FrameNet ontology project. The object movement is build by first extracting the words and enriching them with the semantic analyzer. Then, weight tables, necessary to obtain characteristics values (orientation, shape, trajectory...) for the verb are used in order to get a 3D path, as realist as possible. The weight tables were created to make parallel between features defined for SelfMotion verbal concept (some provided by FrameNet, other determined during the project) and values used in the final algorithm used to create 3D moving representations from input text. We evaluate our analyzer on a corpus of short sentences and presents our results.

Keywords: FrameNet, 3D visualization, animation, SelfMotion, verbal concept, multimodality

1. Introduction

Multimodality can be seen as a way to express textual content in alternate representations like audio or visual animations. Recently, this field of research received an increasing interest from both community of Natural Language Processing (NLP) and Imaging according to its potential use in many user interface oriented applications. As investigated by (Adorni et al., 1984), *the relation between language and space has long been an area of active research*. The multimodality in the context of text conversion into animation or graphical representation have been studied in various way. A milestone was the WordsEye (Coyne and Sproat, 2001) system. It provides a blank slate where the user can paint a picture with words: the description may consist not only of spatial relations, but also actions performed by objects in the scene. In its first evolution the multimodal engine of WordsEye is able to treat a sentence, identify objects in it with their attributes. Like most of those first period applications, the static nature of WordsEye offers no possibilities to treat neither the question of verbal trajectories describing motion. The next generation of such systems tried to introduce animation as a multimodal feature. Carsim (Dupuy et al., 2001; Nugues et al., 2004), is dedicated to the visualization and animation of 3D scenes produced from car accident written reports. E-Hon (Sumi and Nagata, 2006) uses animations to help children to understand a textual content. It provides storytelling in the form of animation and dialogue translated from original text. The text can be a free on-the-fly input from a user. The animation engine of Confucius (Ma, 2006) accepts a semantic representation of texts and uses visual knowledge to generate 3D animations. In a recent work Roth (Roth and Frank, 2009) describes an online application that uses Natural Language Generation (NLG) methods to generate walking directions in combination with dynamic 2D visualization. In this communication, we introduce an algorithm whose purpose is to visually represent the sense of verb related to an object and its derivative representation as an animated sequence in 3D space. We specially want to study the possibility, according to a standard verbal ontology, to build a generic 3D graphical path which allows the representation of verbal concepts in a 3D virtual space. This also implies to identify verbal concepts - according to their meaning in a particular context - that can be linked to an animated 3D graphical representation.

This article is structured as follows : in section 2. we investigate the specific nature of verbal action concepts and their relation with their possible representation in space. In section 3. we present our proposed system to solve this problem. In section 4. we describe the text to trajectory interpreter we developed to transform verbal expression into visual representation. In section 6. we experiment the system using sentences and verify its visual results.

2. Related work

The literature (Ruhlmann et al., 2010; Ma, 2006; Piwek, 2003) generally adopts as a basic architecture model for processing graphical representation of text sentences, a chain of successive modules dedicated to transform texts into a more abstract conceptual representations. This abstract representation is assumed to be easier to transform into spatial graphics primitives.

At the beginning of the pipeline modules, there is a sequence of text objects, organized with a linear sequential structure, and at the end, there is a spatial representation, spatialized, with possibly graphical objects in motion. However, this architecture, commonly accepted by the multi-modal community, is difficult to deploy as the question of text to visual transition face many theoretical issues. In the generic field of graphical representation, (Tversky et al., 2002) claim that correspondences between mental and graphical representations *suggest cognitive correspondences between mental spaces and real ones*. In the perspective of transforming a *conceptual-intention* into a visual representation, (Johnson-Laird, 1998) considers that visual representation of mental models *cannot be reduced to propositional representations* [1] *[as] both are high-level representations necessary to explain thinking.*[2] Johnson-Laird considers also that *mental models themselves may contain elements that cannot be visualized.*
According to this, it appears in the perspective of a text to animation computer application, that the correspondences between semantic abstractions extracted from free text and visual representations are not always relevant to a simple sentence parse and rendering in a graphic engine. In pictorial arts, the correspondences for mental representations allowed by imagination, are obtained by a cognitive transformation of physical laws, natural spaces and transgression of common sense to adapt an animation or a static image to the mental representation. Finally, we can consider that animated results of those specific transformations are equivalent to the creative ones observed in artistic and entertainment applications like computer games, movies, cartoons. This particular aspect of natural language driven image generation and the role of physic limitations has been investigated by (Adorni et al., 1984) who consider that such a cognitive transformation should be relevant to a computer AI problem.
Such pipelined architecture have to deal with specific aspects like common sense interpretation (Liu and Singh, 2004; Lieberman and Liu, 2002) and ontological representations of space according to textual formulation (Bateman et al., 2010).

At the final stage, an important aspect is the exact identification of objects motion, according to the verbs used to describe them, in its context. The potential textual description of a moving object in a space is usually conveyed by a verb, as a word that in syntax conveys an action (bring, read, walk, run, learn). Our system focuses on this final step. It uses Framenet (Baker et al., 1998), an ontology[3] which makes description of Verbal-concept prior to words by associating meaning to a set of potential verbs. The surface form of the verb (the word used to convey its meaning in a textual form) can have a unique sense (e.g. Moving) for different verbs (e.g. Travel, Slide, Move, Displace). Framenet uses various categories of classes called *frames* to categorize verb actions, like `SelfMotion` or `See_through`.
We associate intermediate *look-up tables* to the Framenet Frames to establish the exact semantic relation between a `SelfMotion` verb in a sentence and its representation in 3D graphic space with a drawing function. This extended spatial version of the Framenet is then used by an interpreter to produce animations according to text.

3. Architecture

The pipeline architecture of the proposed system (Figure 1) consists in the following components:

1. The part-of-speech (POS) tagger that assigns its class (verb, adjective, ...) to each word of the sentence,

2. The semantic engine that interprets the logical relation between words in the sentence,

3. The Text-to-trajectory interpreter that assigns a trajectory to a given verbal concept,

4. A Graphic engine that renders the animation according to the information collected in previous steps.

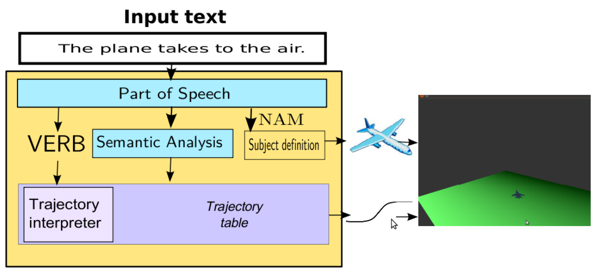


Figure 1: Architecture of the system from the input text to the corresponding animation

The POS tagger comes from Stanford Natural Language Processing tools (NLP) tools. The semantic analyzer, described in (Zouaq and Gagnon, 2010), aims at extracting logical representations from free text based on dependency grammars. When the nature of words (verbs, adjectives, nouns) and the logical relations between them are identified, the text-to-trajectory interpreter tries to define which trajectory is the most accurate to represent the motion.

4. Text-to-trajectory interpreter

The text-to-trajectory interpreter is based on a set of features used as interpretation rules according to a verbal concept. We based our study on the project FrameNet which is a lexical ontology providing lexical units associated to a frame concept. The idea of FrameNet is that words can be understood by associating semantic frame to them. A semantic frame consists in associating words to a set of event, environment, relation and entity.

[1] Defined by (Johnson-Laird, 1998), page 442 as*representations of propositions in a mental language.*

[2] (Johnson-Laird, 1998) page 460.

[3] Available online and for download at `https://framenet.icsi.berkeley.edu/`

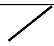
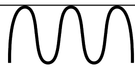
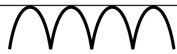
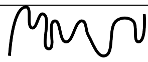
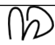
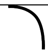
Type	Mathematical primitive	Illustration
rectilinear	$y = ax + b$	
curvilinear	$y = cos(ax)$	
curvilinear	$y = abs(cos(ax))$	
random	$y = rand(x)$	
specific [to vault, to pounce]	$y = ax + b, t \leq 2; y = abs(cos(ax)), t \geq 2$	
specific [to slouch]	$y = rand(t), t \leq 2; y = 0\ otherwise$	
specific [to take to the air]	$y = ax + b$, height raising in time	
specific [to swing]	$(1 - cos(t), t + sin(t)), -3 \leq t \leq 3$	
specific [to slosh]	$(cos(t), sin(t))$	

Table 1: Primitive mathematical functions used to describe the shape of a trajectory

In the present article, we based our application on SelfMotion concept. This concept is defined as: *The Selfmover, a living being, moves under its own direction along a Path.* In FrameNet the concept of SelfMotion is represented as a frame to whom are linked frame elements also called features. For SelfMotion concept, FrameNet associates four principal frame elements: Direction, Goal, Path and Source. Words that evoke this frame (*to walk*, *to run*, *to jump*, *to swing*) are called lexical units of the SelfMotion frame.
Based on this work we started our development by characterizing each verbal concept with frame elements. For example, run is associated to an advance direction, forward goal, rectilinear path and actual place of the entity source. The features used as weight tables for each verbs were improved by iterations. To make differences between verbs having same characteristics, for example *to walk* and *to run* had same characteristics with the fourth first features (direction, goal, path, source), we determine a speed feature, useful to distinguish between the two verbal concept. Height weighting was necessary to distinguish lexical units as *to fly* or *to crawl*. Gradually we were able to construct our correspondence tables used next to build our 3D representation. The more features we had, the more accurate was the visualization.
The text-to-trajectory interpreter aims at constructing a 3D representation using the following correspondence table.

1. Orientation of a trajectory
2. Shape of a trajectory
3. Speed along a trajectory
4. Speed modulation according to adjectives
5. Amplitude and height modulation

4.1. Orientation of a trajectory

We first defined a direction to each verbal concept as in Table 2. Our experiment shows that three different major orientations are required to distinguish between the possible orientations of the verbal concept: forward movement, backward movement, and stationary movement. Stationary movement makes use to verb as *to dance*, *to jump*, *to swing* since the entity is moving around the initial position.

Orientation	Examples
forward	*advance, move forward*
backward	*go back, retreat*
stationary	*swing, dance*

Table 2: Values used to describe the orientation of a trajectory

4.2. Shape of a trajectory

To describe the shape of a trajectory, we defined four basic types: *rectilinear*, *curvilinear*, *random* and *specific*. In our purely spatial approach these shapes need to be defined by mathematical functions. By combining two mathematical primitives, one horizontal (i.e. at ground level, in some direction) and one vertical, we can construct a quite realistic spatial trajectory. According to our preliminary tests, we found that only three basic functions and seven specific functions, shown in Table 1 are enough to characterize the 136 lexical units contained in the SelfMotion frame.
Due to different mental notion of trajectories (the main shape associated to verbal concept was judged by a single person), a secondary shape was also incorporated to the general table to let a second possible choice to represent the trajectory. By default, the trajectory interpreter choose the

Adjective:	*fast*	*slow*	*calm*	*hurry*	*quick*	*rapid*	*haste*	*rush*	*accelerate*	*swiftly*	*racing*
Factor:	2	0.5	3	2	2	2	2	2	11	12	13

Table 3: Speed modulation according to adjectives

main shape for the visualization. Some verbs like *to vault, to pounce, to slouch, to take to the air, to swing, to stroll* require more specific mathematical function. For example, *to swing* uses the parametric equation of a circle while *to take to the air* uses the rectilinear function with a height increasing in time.

One has to notice the very few mathematical functions actually required by our algorithm: only three basic mathematical functions are enough to represent almost the 136 `SelfMotion` lexical unit listed by FrameNet. Six more specific functions were required to give better and more realistic trajectory to lexical units such as *to slouch, to take to the air* or *to vault*.

4.3. Speed along a trajectory

At this point, we can observe that two verbs can share the same trajectories: e.g. *to walk* and *to run* share the same trajectories, the difference coming from speed. Therefore, a speed weight was deemed necessary to distinguish between them. We rated each verbal concept with a speed between 0 (static) and 10 (moving very quickly), as shown in Table 4.

Speed	Description	Example
10	speed of a plane or a flying entity	*The jet is flying*
9	car or motorcycle speed	*The vehicle is coasting*
8	horse speed	*The horse moves at a canter*
7	running or hurrying	*The man is flouncing*
6	moving quickly	*The bee darted across the room*
5	usual walking speed with no hurry but not in a leisure way either	*People are walking in the street*
4	walking leisurely or quietly and carefully	*He gallivants around the town*
3	walking or moving wearily, reluctantly or aimlessly	*He trapes during all the stroll*
2	climbing or moving in an awkward and laborious way	*The sportsman clambers with difficulty*
1	moving in a very slow way, sleepwalking	*He sleepwalks every night*
0	not moving	*He stands up looking around him*

Table 4: Speed along a trajectory

4.4. Speed modulation according to adjectives

We included a speed modulation table that associates speed adjectives (such as *faster, hurried, slowly, quietly*) with a modulation factor shown in Table 3. This rate modulates the characteristic speed contained in the speed attribute of each lexical unit. This treatment was taken into consideration as a proof of concept for attribute impact in the resulting 3D graphical animation.

4.5. Amplitude and height modulation

Finally, we defined a set of modulation criterion necessary to obtain even more realistic trajectories. As soon as we increased the granularity of our model, we started to obtain adequate trajectories allowing distinction between verbs like *to bustle* or *to caper*. The amplitude modulation is similar to some wriggle factor and is used for lexical units having curvilinear or random trajectories. The amplitude feature was used as a stick to represent the uncertainty of a path and to modulate the random function used for unpredictable trajectory. We were able to distinguish between verbs like *to lurch, to meander* or *to wriggle* using coefficient going from 0 (no modulation) to 9 (maximal amplitude) as shown in Table 5.

Index	Description	Example
9	vaulting or having a curvilinear trajectory	*The sportsman is propelling himself with a pole*
8	taking to the air	*The airplane begins to fly*
5	wandering aimlessly	*The man was wandering in the hills*
4	swinging or having a winding path	*The children are swinging in the playground*
3	moving about stealthily or restlessly	*Somebody was creeping upstairs in the house*
2	staggering or lurching	*He was walking tipsily*
1	random trajectory but in a hurrying way	*He storms out (of the room)*
0	no range	*He walks straight on*

Table 5: Horizontal amplitude modulation

We also incorporate a height feature useful to distinguish between lexical units having different height. *To crawl* and *to walk* are distinct in term of height criteria. So, we add a height scale shown in Table 7, going from -1 (the entity height is decreasing in time) to 11 (the entity height is increasing in time), 3 (height of a walking man) was the reference.

SelfMotion		Main Shape		Secondary Shape			Amplitude modulation	
Verbal-concept	Orientation	Horizontal	Vertical	Horizontal	Vertical	Speed	Horizontal	Vertical
advance	forward	rectilinear	-	random	-	5	0	3
amble	forward	rectilinear	-	random	-	4	0	3
back	backward	rectilinear	-	-	-	5	0	3
barge	forward	rectilinear	-	-	-	7	0	3
bop	forward	rectilinear	-	random	random	6	0	3
bound	forward	rectilinear	curvilinear	random	random	7	3	11
burrow	forward	rectilinear	rectilinear	random	-	4	0	-1
bustle	forward	random	-	rectilinear	-	7	5	3
canter	forward	rectilinear	curvilinear	random	-	8	3	4
caper	forward	random	-	rectilinear	-	6	2	3
clamber	forward	-	random	-	rectilinear	2	2	11
climb	forward	-	random	-	rectilinear	4	2	11
clomp	forward	rectilinear	-	random	-	4	0	3
coast	forward	rectilinear	-	-	-	3	0	3

Table 6: Samples from the global correspondence table used by the interpreter.

Height	Description	Example
11	height increasing in time	*The man is climbing*
10	fly in the sky	*The plane is flying*
5	walking by bounding, making leap	*The kangaroo is bounding*
4	person on motorbike, riding a horse	*The man is on a motorbike*
3	adult walking on ground	*He is standing up*
2	adult walking with a stoop	*He stoops in order to make himself look shorter*
1	grazing the ground	*The bird skims on the surface of the lake*
0	crawling or lying on the ground	*The man is swimming at the surface of the water*
-1	digging a hole, height decreasing in time	*He is burrowing a hole in the ground*

Table 7: Vertical amplitude modulation (height)

5. Software implementation

Using those interpretation features, we built a unique table called Verb to Motion Table (VMT) containing all `SelfMotion` verbal concepts associated to frame elements. In this table, each motion verbal concept according to the FrameNet ontology is described with at least one specific characteristic movement definition. An sample of this table is shown in Table 6. The interpretation process will consist in determining what entry from the VMT, according to the motion detected in the sentence, will be used to generate the graphic animation. This is done through a set of rules that find the correspondence between a lexical unit from FrameNet matching the verb surface form in the sentence. If this lexical unit matches more than one verbal-Concept of FrameNet, a disambiguation process is applied. This process evaluates the similarity between words contained in the FrameNet sentences samples related to each Verbal-Concepts and the words from the sentence to analyze.

We implemented this global table in a 3D animation software using OpenGL[4]. This software prototype can receive a series of short sentences and process them. For each analyzed sentence, the global table is searched to obtain the characteristics of the related trajectory, to allocate the right feature values to the entity and finally using them into 3D spatial visualization.

6. Evaluation and experiments

The goal of our experiments was to evaluate the accuracy of trajectory selected by our software prototype according to a given sentence. As our software is not restricted to valid semantic productions that do not violate commonsense and physical laws (see (Charton et al., 2010) for discussion about the commonsense problem), we did not focus on entity realism: subjects of sentences where symbolized as a cube when no drawing model was available for representation (see Figure 3).

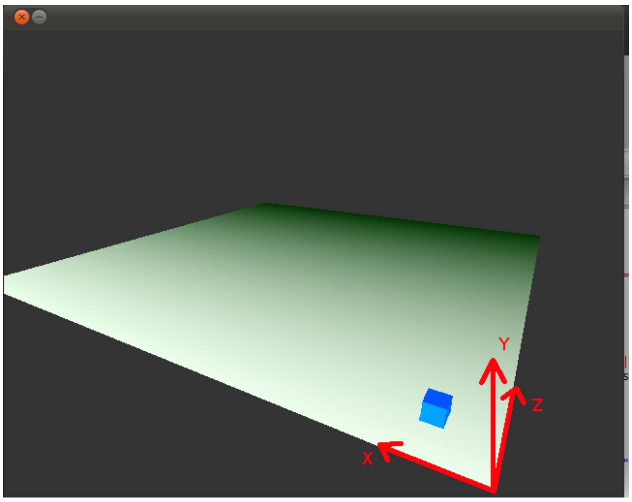


Figure 3: Base tray in three-dimensional mark

[4]See `http://www.opengl.org/`.

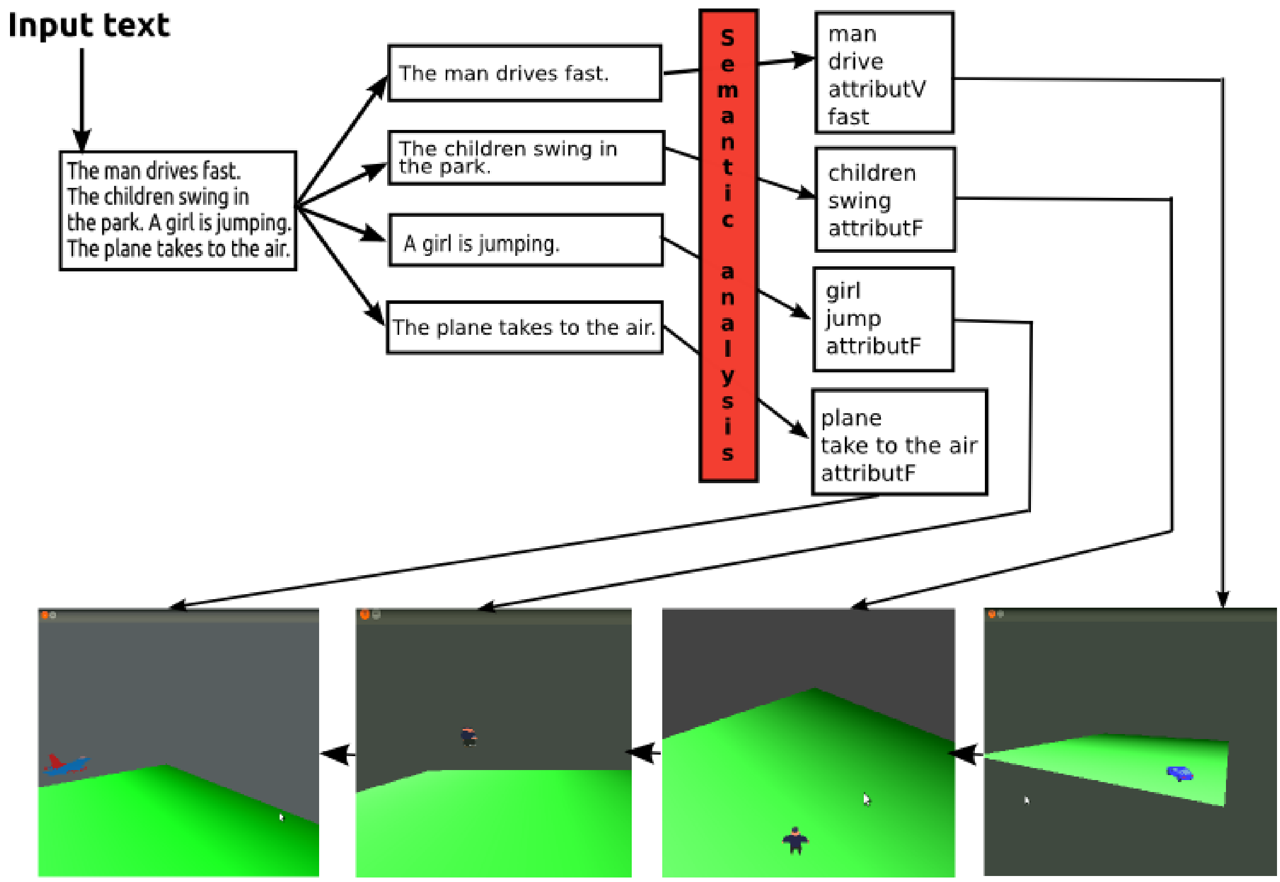


Figure 2: Samples of interpretation and their result as visualization.

Then we improved our model by adding some basic 3D objects in the scene in order to obtain more realistic representation (see Figure 2). The evaluation has been conducted manually. Each sentence has been launched one time in order to obtain the corresponding 3D animation and comparison was made with the sentence provided as input to the algorithm. We only evaluated the accuracy of the taken path, and not the environment and object used in the representation since it was not the goal of our study.

To validate the architecture of our system we proceeded in two steps:

1. First, we submitted a small group of single sentences to the interpreter and then adjust the values in tables (Table 6).

2. Then, we submitted the full set of sentences contained in corpus to the system, and evaluated visually the accuracy of the animation.

The corpus used for development and testing was built manually and contains 400 sentences made of a minimum of one subject and a `SelfMotion` verb. To build this corpus, we selected verbal-concepts described by Framenet `SelfMotion` and associated each of them with a subject. For example, the subject can be referred to a person (*she*, *he*) or an object (*it*, *the plane*). We also treated plural cases: when the sentence subject was in plural form, like *men*, the animated entity corresponding is a group of several man objects.

This test corpus was divided in two sets of 100 and 300 sentences. The first set of 100 sentences was submitted to the system and used to adjust values in the tables of the text-to-trajectory interpreter. Then, we submitted the second set of 300 sentences to the system and evaluated visually the accuracy of the animation. We considered a sentence as successfully represented when its trajectory was relevant and took the proper path according to a commonsense interpretation of the sentence acknowledged by the two evaluators.

6.1. Results.

Around 70 out of 300 were classified as *failure* since the produced trajectory was not relevant. Errors of representations mainly happened for verbs which had specific complex trajectories such as *to slosh*, *to vault*, *to take to the air*.

It appears that verbal concepts using random shape, horizontally or vertically, were source of failure. Actually, aleatory trajectories (like *to dance*, *to meander*, *to bustle*) were more difficult to represent using the VTM table. This kind of verbal concept needs some random part which can not be based on the random function available in the Math C++ API. In order to be more realistic, it may be a good idea to use a personalized random function for this project. The VTM table was generated manually. The values used in this table come from the different weight tables defined

Test sentence	Result	Explanation
The man walks	Success	Right trajectory
She dance	Success	No trajectory, random movement with right range (not too high but enough to represent dance)
Plane takes to the air	Failure	Not significant path, height not increasing uniformly
Child is swinging	Failure	Move just back and not forth, no repetition to make the usual balancing movement
He is climbing	Success	Random vertical trajectory Then, we used more complicated sentences and text to validate the system., increasing in time
He bounds	Success	Significant trajectory with object making bound on the ground without going under the tray
He vaults	Failure	Only curve trajectory without right trajectory representing running preceding jump
He drives. The woman walks one the sidewalk	Success	Succession of two displacements, first showing drive trajectory and then man walking
They meander and stroll around town	Failure	Speed too high to show the leisure race

Table 8: Sample of sentences from the test corpus, results of our interpreter and comment about this result.

in the project. This values were evaluated manually, by one person, which implies a necessary correction. Having only one person to evaluate the verbal concept and to attribute a value can be source of mistake. The perception of each verbal concept is not the same in all human mind which means we need to define a more rigorous method to construct the different weight tables. There are lot of verbal concepts which implies that some verbal concept may have been treated quickly and without same consideration as another verbal concept. We provide an extract of those results in Table 8.

7. Conclusions and perspectives

In this paper, we presented a multimodal system dedicated to animated representation of verbal concept contained in a text, using the Framenet ontology. Our results show that a motion related verbal concept can be associated to a combination of basic trajectories to automatically produce a 3D animation. We validated the principle of the association of trajectories to motion verbs to produce a realistic spatial representation of a text sentence. This approach can open new possibilities to create novel architecture dedicated to animated graphic representation of text. We plan to apply this system to open corpus, not limited to representable sentences and to include disambiguation techniques in the interpretation process. We specially want to focus on disambiguation of metaphoric texts and common sense validity.

Reproducibility

The software code, developed in C++, the corpus, the visualization objects and the interpretation tables are released publicly at the following url: `http://gitan.polymtl.ca`.

8. References

Adorni, G., Di Manzo, M., and Giunchiglia, F. (1984). Natural language driven image generation. In *Proceedings of COLING*, pages 495–500, Hillsdale, NJ. 10th International Conference on Computational Linguistics and 22nd annual meeting on Association for Computational Linguistics.

Baker, Collin F., Fillmore, Charles J., and Lowe, John B. (1998). The berkeley framenet project. *Proceedings of the 36th annual meeting on Association for Computational Linguistics* -, page 86.

Bateman, J.A., Hois, Joana, Ross, Robert, and Tenbrink, Thora. (2010). A linguistic ontology of space for natural language processing. *Artificial Intelligence*, jun.

Charton, Eric, Gagnon, Michel, and Ozell, Benoit. (2010). A preprocessing system to include imaginative animations according to text in educational applications. In *Mog 2010*, Dublin.

Coyne, B. and Sproat, R. (2001). Wordseye: An automatic text-to-scene conversion system. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, volume ternetchap, pages 487–496. ACM New York, NY, USA.

Dupuy, Sylvain, Egges, Arjan, Legendre, Vincent, and Nugues, Pierre. (2001). Generating a 3d simulation of a car accident from a written description in natural language. *Proceedings of the workshop on Temporal and spatial information processing* -, pages 1–8.

Johnson-Laird, PN. (1998). Imagery, visualization, and thinking. *Perception and Cognition at Century's End*, pages 441–467.

Lieberman, Henry and Liu, Hugo. (2002). Adaptive linking between text and photos using common sense reasoning. *Science*, pages 2–11.

Liu, Hugo and Singh, Push. (2004). Commonsense reasoning in and over natural language. In *Knowledge-Based Intelligent Information and Engineering Systems*, pages 293–306. Springer.

Ma, M. (2006). *Automatic conversion of natural language to 3D animation*. Thesis, University of Ulster, Faculty of Engineering.

Nugues, Pierre, Dupuy, Sylvain, and Egges, A. (2004). Carsim: un système pour convertir des textes en scènes tridimensionnelles animées. *14e congrès de l'AFRIF-AFIA, Reconnaissance des formes et Intelligence artificielle, Toulouse,*.

Piwek, P. (2003). A flexible pragmatics-driven language generator for animated agents. In *Proc. of EACL*, page 151, Morristown, NJ, USA. Association for Computational Linguistics.

Roth, M. and Frank, A. (2009). A nlg-based application for walking directions. In *Companion Volume to the Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, Singapore*, pages 2–7.

Ruhlmann, Laurent, Ozell, Benoit, Gagnon, Michel, Bourgoin, Steve, and Charton, Eric. (2010). Improving communication using 3d animation. In *KES2010 14th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems*, Cardiff.

Sumi, Kaoru and Nagata, Mizue. (2006). Animated storytelling system via text. In *Proceedings of the 2006 ACM SIGCHI international conference on Advances in computer entertainment technology*, volume Proceeding. ACM SIGCHI international conference on Advances in computer entertainment technology.

Tversky, Barbara, Morrison, J.B., and Betrancourt, Mireille. (2002). Animation: can it facilitate? *International Journal of Human-Computer Studies*, 57(4):247–262.

Zouaq, Amal and Gagnon, Michel. (2010). Semantic analysis using dependency-based grammars and upper-level ontologies. In *CICLING 2010*. CICLING.