Mechanisms for Multimodality: Taking Fiction to Another Dimension

Kevin Glass *
Department of Computer Science
Rhodes University
Grahamstown, South Africa

Shaun Bangay †
Department of Computer Science
Rhodes University
Grahamstown, South Africa

Bruce Alcock ‡
Department of Computer Science
Rhodes University
Grahamstown, South Africa

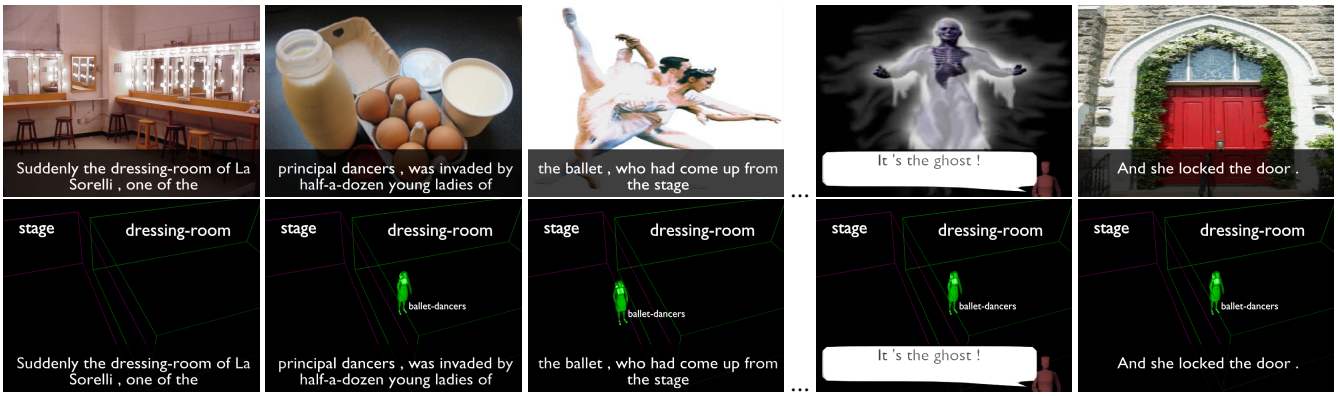


Figure 1: Alternative fiction representations using subtitles enhanced with avatars, image-based graphics and 3D graphics. Text sourced from *The Phantom of the Opera* by Gaston LeRoux. Image sources are listed at the end of the paper.

Abstract

We present methods for automatically constructing representations of fiction books in a range of modalities: audibly, graphically and as 3D virtual environments. The correspondence between the sequential ordering of events against the order of events presented in the text is used to correctly resolve the dynamic interactions for each representation. Synthesised audio created from the fiction text is used to calibrate the base time-line against which the other forms of media are correctly aligned. The audio stream is based on speech synthesis using the text of the book, and is enhanced using distinct voices for the different characters in a book. Sound effects are included automatically. The graphical representation represents the text (as subtitles), identifies active characters and provides visual feedback of the content of the story. Dynamic virtual environments conform to the constraints implied by the story, and are used as a source of further visual content. These representations are all aligned to a common time-line, and combined using sequencing facilities to provide a multimodal version of the original text.

CR Categories: I.3.7 [Computer Graphics]: Three-dimensional graphics and realism;1.2.7 [Artificial Intelligence]: Natural Language Processing—Language parsing and understanding

Keywords: multimodality, text-to-scene conversion, constraint solving

*email: k.glass@ru.ac.za
†email: s.bangay@ru.ac.za
‡email: bruce.alcock@gmail.com

Afrigraph 2007, Grahamstown, South Africa, October 29–31, 2007.
© 2007 ACM 978-1-59593-906-7/07/0010 $5.00

1 Introduction

1.1 Problem Statement

The conversion of fiction text to an unified multimodal representation is a difficult and subjective process which requires substantial human intervention at every step of the process (as can be indicated by the lengthy list of credits at the end of any movie). Some steps in the process stand to benefit from automation, freeing the human component to concentrate on the creative aspects. We investigate a repertoire of techniques for producing multimodal representations from unrestricted fiction text. Applications of these techniques includes producing audio versions of the books, creating movies from the book, and even producing interactive virtual environments in which the story can unfold. In all of these scenarios, the sequencing and synchronization of the different media is vital.

1.2 Background

This context of this research is a Text-to-Scene conversion system, which has the goal of converting descriptions found in fiction books to graphical formats. Fiction text as written for human consumption describes interactions between characters, objects and locations over a period of time in a fictional world. This virtual world orders events sequentially (unlike the text which may contain flashbacks, or descriptions of concurrent events) and so requires its own time-line which is independent of the order in which these events may be presented. Graphical output based on the virtual world must be correlated to the timing of the actual text if it is to be used to support presentation.

Techniques and technologies for extracting the required semantic information from fiction texts has been reported previously, and so is beyond the scope of this paper. The process involves analysing the structure of the text using parts-of-speech taggers and syntactic parsers [Glass and Bangay 2005], and using a rule-based learn-

ing mechanism for automatically annotating the text with relevant information [Glass and Bangay 2006]. The information used by the techniques described in this paper is extracted from automatically generated and manually created annotations. The values used include: entities, specifically in the form of avatars, objects and areas described in the text; quoted-speech and the avatar responsible for the speech; characteristics of the avatar including gender; noun phrases for locating audio foleys and background images; and spatial relations between entities described using preposition-based descriptions.

1.3 Overview

Techniques for synchronising the different representations of fiction texts are fundamental to all processes described in this paper, and are covered in Section 3. Strategies for producing each output format are discussed in turn: the creation of audio representations is described (Section 4), followed by descriptions of graphical representations, including enhanced subtitles (Section 5), image-based representations (Section 6), and 3D virtual environment representations of the fiction text (Section 7). The process of sequencing the automatically generated, synchronised multiple media segments (Section 8) demonstrates the integration of all of these representations.

2 Related Work

Alternative representations for fiction books range from digital representations of actual books [Chu et al. 2003] and providing ambient sounds effects [Back et al.], to virtual interactive representations [Billinghurst et al. 2001]. However, in all of these approaches the non-textual representations are generated manually, while interaction with the book is the primary focus.

Efforts have been made for the automatic conversion of fiction stories to alternative representations. The idea of creating enhanced audio versions of a fiction story has been previously explored, specifically with the aim of using different voices for each character [Glass and Bangay 2006; Zhang et al. 2003]. Graphical storytellers have also been used in conjunction with synthesised speech to enhance the storytelling experience [Piesk and Trogemann 1997]. We adopt a simpler subtitling approach to aid in comprehension, where the speaking characters are explicitly marked using graphical avatars and speech-bubbles.

Several variations on the process of creating graphical alternatives to fiction text automatically have been explored. Image based representation of stories is performed by the Story Picturing Engine [Joshi et al. 2004], which provides images corresponding to portions of text. These images are sourced from a database of well annotated images. Projects such as CONFUCIOUS [Ma 2002] and SWAN [Lu and Zhang 2002] create three dimensional animations from fiction stories, but require that the input language be restricted to simple, well structured forms. Systems such as 3DSV [Zeng et al. 2003; Zeng et al. 2005] visualise natural language stories, but do not present any examples using unrestricted text, nor do they handle the aspect of time. CarSim [Akerberg et al. 2003; Tabordet et al. 1999] creates 3D visualisations of written road accident reports, but is limited to only this domain. Other relevant examples of the interpretation of natural language is for the placement of objects in a static scene [Clay and Wilhelms 1996], creating still 3D images using natural language [Coyne and Sproat 2001], and the use of natural language for controlling agents in a virtual world [Badler et al. 2000; Bindiganavale et al. 2000].

The techniques described in this paper represent alternative strategies devised to achieve these goals. Our most significant contributions involve the methods used to synchronize and sequence the output from each strategy, and extract timing information from the different media. In particular, the use of audio versions of text to derive timing information is unique to this field.

3 Alignment of multimodal representations

Before introducing the mechanisms involved in creating multimodal representations, we first explain the manner in which the media are synchronized to form a coherent output. The paradigm is based on a typical film sequencer which is able to handle multiple tracks of audio and video. Figure 2 illustrates the various forms of media used to represent the fiction book.

The order and duration of events defined by the fiction text is known as the *presentation time-line*. Many forms of output (such as audio books or movies) use this ordering of events, and so each type of media produced must be capable of being sequenced relative to this time-line.

Each audio file, image or animation inserted into the sequencer is represented by a *segment*, which indicates the starting time and duration that the corresponding media will have in the final output. Our aim is to automatically produce a number of different types of segments, placing them in the correct order in the sequencer to match the fiction text. This requires a technique that is capable of deriving starting times and durations from the text for each segment.

Audio segments within the presentation time-line are generated automatically from segments of text using a text-to-speech synthesiser, producing audio files. The duration of each segment is determined by the length of its corresponding audio file. When placed in order as specified by the book the absolute start time for each segment can be calculated relative to the start time and duration of the preceding segments, as indicated in Figure 2.

Every segment originates due to a token sequence in the original text, which we refer to as the *textual trigger*. Triggers may be entire sentences (for example, in audio and subtitle segments in Figure 2) or a group of consecutive tokens (image segments in Figure 2). These triggers are used to derive absolute starting times for each new segment using the audio segment in which the first token of a trigger occurs.

The starting time of the trigger relative to the audio segment is approximated as an offset based on the location of the trigger in the segment:

$$offset_{trigger} = \frac{position\,of\,token_{text\,segment}}{number\,of\,tokens_{text\,segment}} * audio\,length$$

$$start\,time_{trigger} = start\,time_{audio\,segment} + offset_{trigger}$$

This process, for example, is used to derive the starting time for the image segment corresponding to the token “legs” in Figure 2.

Durations of segments are derived according to the type of segment being created. In some instances the length of the audio version of the trigger is used as a duration while in other instances, a segment may continue until another segment is placed in the same channel. Examples of these include subtitles and images respectively, indicated in Figure 2.

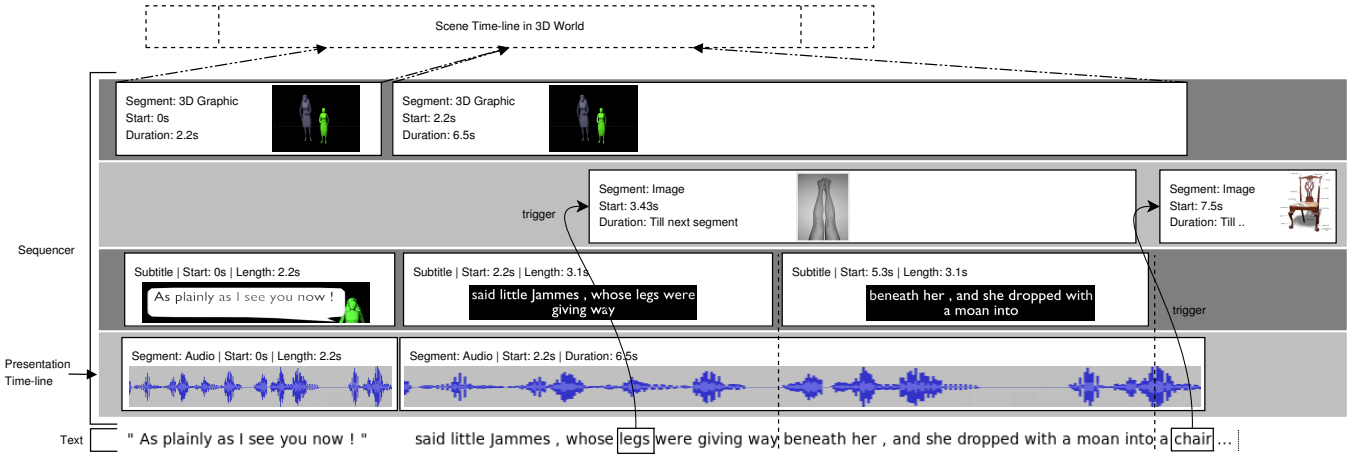


Figure 2: Illustration of the sequencer-based approach to combining alternative forms of the input text, and aligning them according to an audio-based presentation time-line.

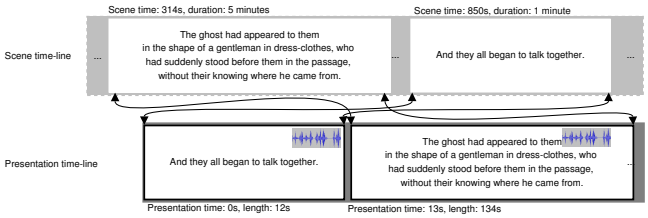


Figure 3: Example of the relationship between presentation time and scene time.

Scene Time-line We employ a second representation of time which we call the *scene time-line,* indicated in Figure 2. The scene time-line represents sequences of events as they would occur in the virtual world described in the fiction text. Segments which may be classified as "flash-backs" in the presentation time-line are placed in the correct order in the scene time-line. This allows for the construction of a time-based 3D scene which accurately reflects the state of the virtual world at any point. When a segment of the presentation time-line must be rendered the correct portion of the 3D scene is located from the scene time-line. The events of interest are metaphorically "filmed" creating a segment of animation, and this segment is placed at the correct location in the sequencer.

Figure 3 illustrates the relationship between the presentation time-line and the scene time-line. Notice that the second portion of text "The ghost had appeared...", which describes actions that occur in the past, occurs first in the correct chronological order with respect to the virtual world being described. Each segment in the scene time-line is attributed with a start time and a duration. Currently these are manually specified.

4 Strategies for audio

We describe two options for creating audio representations of fiction text. The first is the generation of an audio version of the narrative using a text-to-speech synthesiser, while the second is the inclusion of the sound effects, or foley, described in the text.

4.1 Audio Narration

Audio versions of text are easily generated using standard text-to-speech technology, such as eSpeak (`http://espeak.sourceforge.net/`). However, the speech generated is often monotonous and difficult to follow. While improvement in the prosody of the speech synthesizer can address this to an extent, we believe that this problem may be further alleviated by choosing voices appropriately for each character in the story. This adds an element of variation to the audio narrative.

Fiction text contains quoted text representing speech emitted by different characters in the story and each character requires a different voice. Most speech synthesizers provide a limited number of voices, usually including a male and female voice. While entirely new voices can be created, experience has shown this to be a painstaking and time consuming process [Hood 2004]. Instead we create a range of voices by choosing a different pitch for each character, within a range that is still appropriate to the specified gender. Other parameters may also be modified according to the abilities of the speech synthesizer being used. Each section of quoted speech is rendered with the appropriate voice, and recorded to an audio file. Non-quoted speech is rendered using a default *narrator* voice.

The characters that participate in the story must be identified along with their gender so that we can select either a male or a female derived voice. Each instance of quoted speech must be extracted from the text, and the correct speaker for the quote accurately identified. This process described in detail in previous work [Glass and Bangay 2006].

The use of avatar specific voices adds valuable variation to the narrated output, and allows avatars to be identified by the listener even in dialog where they are not explicitly named. In practice automatic audio renderings alone are not sufficient for ready comprehension of a fiction story. Visual cues aid in conveying the story effectively.

4.2 Foley

Sound effects are often described in fiction text, illustrated in Figure 4. Nouns such as "rustling" are automatically identified by examining the hypernym (abstraction) tree of the word provided by the WordNet lexical database [Fellbaum 1998]. Any word related to "sound" is identified as a foley, and a corresponding audio

```
Everybody seemed to hear a   rustling   outside the
door.
```

Figure 4: Example foley description from *The Phantom of the Opera* by Gaston LeRoux.

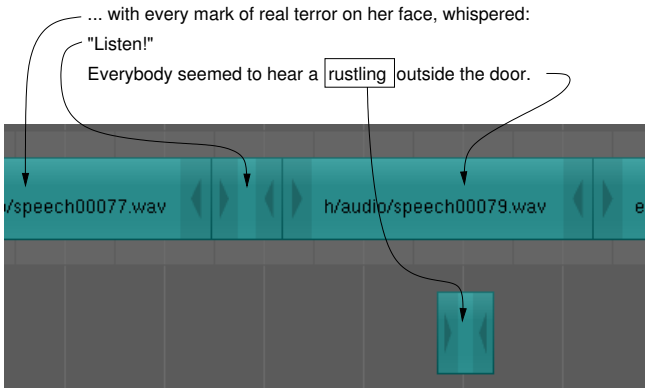


Figure 5: Example sequencing of audio narrations and foleys.

file matching the word must be found. Currently this is done by matching the name of an audio file with the word extracted from the text. In future we plan to create a foley database where each audio file contains descriptive tags for matching, and include the use of Internet-based search engines for locating foleys for obscure terms.

Figure 5 shows a sample of sequenced audio from the *Phantom of the Opera* by Gaston LeRoux. Note that narrated audio is placed into a single channel in the sequencer following the same order as the original text. The foley corresponding to the word "rustle" is inserted into a separate channel, and is mixed into the final rendering at the correct location.

Sound effects further improve the quality of the audio output by adding richness to the output (and an element of humour when a sample that is less than appropriate is selected).

5 Strategies for subtitles

The problem of mispronunciation in synthesised speech can be addressed by providing concurrent subtitles. The text for each segment is rendered as an image which is overlayed above the currently displayed graphics. The times at which subtitles appear and disappear are automatically derived from the location of the corresponding audio files in the presentation time-line.

Subtitles can be further enhanced to provide the appearance of comic style visuals by wrapping the text in a speech-bubble whenever an audio file associated with quoted speech is reached, an example of which is presented in Figure 6. A graphical representation of the appropriate avatar provides a visual identifier unique to the talking character. The process of creating this representation is described in Section 7.3.

The visual component of the subtitles compensates for many of the deficiencies in the quality of the rendered speech, and could also serve to benefit viewers with hearing deficiencies. The use of a visual avatar provides for easier identification of the speaking character, and provides a visual mechanism for differentiating between narration and speech.

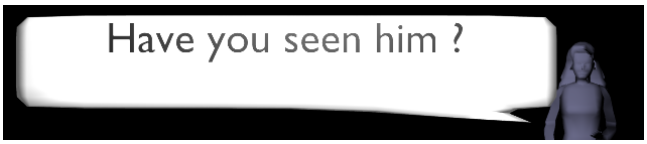


Figure 6: Example subtitle with speech bubble, representing a quote from the character *Sorelli*.

6 Strategies for background images

In addition to subtitles, we present a static image which is used to provide a graphical interpretation of the scene being described in the text. These can be used as a backdrop against which foreground action can take place, or as a source of imagery to further enhance the presentation. This strategy, of displaying images that reflect the content described in the narration is inspired by the Story Picturing Engine [Joshi et al. 2004]. The ready availability of image search engines such as Google Images (`http://images.google.com`) allows us to source images from the Internet using phrases found in the fiction text. We extract all noun phrases using an automatic English parser [Tapanainen and Järvinen 1997], and use these as search criteria for locating images. Similarly to [Joshi et al. 2004] we remove *stop-words,* which include words of type article (the, a), pronoun (he, she, it) and conjunction (but, and) that may be included in a noun phrase.

Searches typically return a number of different images. We download the top five images for each search term and store them in a local cache. These are indexed by a library of search terms which links each term to the set of corresponding images. This library is initially checked for terms matching the extracted noun phrase, and if a match occurs then one of the corresponding images is chosen to be displayed. If there is no match, then the search terms are added to a queue for future download.

Images are placed into the sequencer according to the location of the source text in the presentation time-line, as indicated in Figure 2. The start-time for each image is set to correspond to the time of the first token of the trigger, and the end time set to the start time of the next image. This ensures that the images correspond to the audio and visual subtitles.

Limiting the search to five images allows a degree of coherence in the presentation, in that the same image may reoccur when terms are repeated. The availability of alternatives prevents the content becoming repetitive and predictable. Having alternatives also allows the occasional inappropriate image to be manually culled, although this is rarely required in practice.

Figure 7 presents example images returned for the indicated search terms extracted from *The Famous Five: Five on a Treasure Island* by Enid Blyton. We observe that in many cases the images returned are good representations of the textual descriptions, as in Figure 7(a). The process also produces interesting images for abstract concepts, such as *time* as indicated in Figure 7(b), but whether such images are entirely correct or relevant is unclear. The major pitfall of this approach is indicated in Figure 7(c), where the image annotation matches the search terms, but is used in a very different context to the story. Other scenarios include incorrectly tagged noun phrases due to limitations of the tagger and poor image annotation due to limitations of the search engine.

We also observe that the duration with which each image is displayed must be chosen carefully. In some instances, a sequence of noun phrases occurs one after the next in quick succession, resulting in a number of consecutive images being displayed for very

short periods of time. This can make it difficult to interpret the association between image and text. It does have the advantage that viewers notice different images each time they watch the sequence, making each screening a unique experience.

Sequences of static images lack continuity and introduce substantial context switching overhead for the reader. Viewers have commented favorably on the enhanced reading experience, in particular enjoying the process of matching picture to the text. Continuity and dynamics must be provided by representing the story with automatically generated three-dimensional graphics.

7 Strategies for virtual worlds

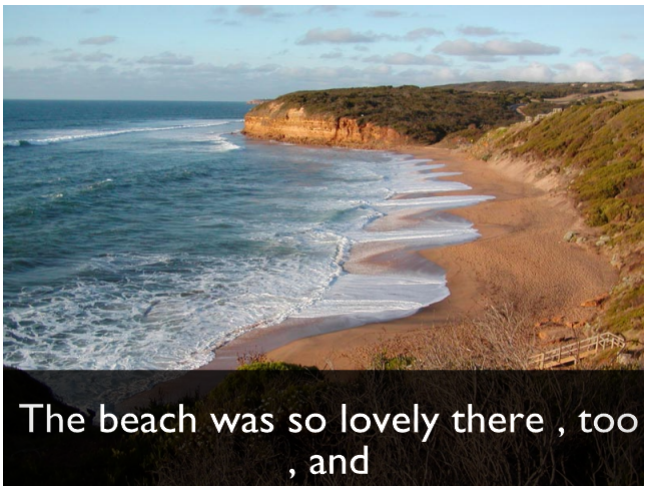
A graphical representation of a book can best be created by extracting a description of the underlying virtual universe. The inhabitants of this universe can be animated to match the requirements of the story and rendered to produce static images or animated sequences that visually represent the text. We define the concept of an *entity* which refers to the principle visual objects in the world: either an area, a character in the story (avatar), or object described in the story. Each of these occur many times in the text, using different descriptions that all ultimately reference to the same entity. Entities are extracted from the text and used to populate virtual worlds. For animation and rendering purposes, each instance is attributed with information such as a unique geometry file, colour, trajectory, and voice parameters. We describe instantiation of the entities in this section; extraction techniques are covered in other work [Glass and Bangay 2006].

Given that a set of avatars, areas and objects have been extracted from a fiction text, these objects must be placed at the correct position in a 3D virtual world, and must move around the world in a manner which reflects the descriptions in the text. We use an approach that specifies the relations that exist between entities at certain points in the books, and transform these relations into mathematical constraints, which when solved, ensure that each entity is at the correct location at every instance of scene-time in the virtual world.

7.1 Constraint-based scene specification

We use a time-based mechanism for specifying the trajectories of entities in a 3D scene, based on the spatial relations between entities specified by the text. The temporal nature of each relation (that is, when it begins and ends) is derived by examining where the trigger occurs in the presentation time-line. This is converted to the scene time-line using the relationship illustrated in Figure 3. All trajectories describe the dynamics of objects in the virtual world and so starting times and durations for the relation are required in scene time. The result is a relation specified to hold over a scene-time span, and we call this an *abstract constraint,* an example of which is presented in Figure 8.

Constraints are transformed into mathematical expressions, which constrain the possible trajectories of each entity. Trajectories, denoted as $R(t)$, are represented using a curve as a function of time t; for example using a spline of n degrees. We define the scene in three dimensions, so the trajectory for each entity is actually composed of three component trajectories; one for each dimension. For example, an entity A has three trajectories $R_x^A(t)$, $R_y^A(t)$ and $R_z^A(t)$. The trajectory provides vectors that define its forward facing direction $\bar{f}(t)$, upward direction $\bar{u}(t)$, and rightward direction vectors

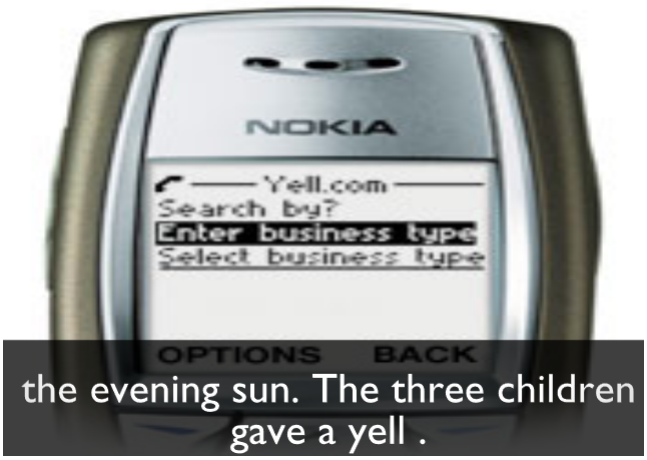


(a) Search term: "beach"

`http://www.surf-blog.net/images/bells-beach/bells-beach-surfing.jpg`

(b) Search term: "time"

`http://www.pbs.org/wgbh/nova/time/images/clocks.jpg`



(c) Search term: "yell"

`http://www.yell.com/img/mobile_image_wap.jpg`

Figure 7: Example images fetched using noun phrases extracted from *The Famous Five* by Enid Blyton.

```
CONSTRAINT:
  Subject:          ballet-dancers
  Relation:         Inside
  Object:           stage
  Start-time:       0.0
  End-time:         0.2
```

Figure 8: Example abstract constraint.

Relation	Function $f(A,B) =$	Constraint
A `InsideOf` B	$\|\|\bar{d}(A,B)\|\|^2$	$f(A,B) < (c^B)^2$
A `AdjacentTo` B	$\|\|\bar{d}(A,B)\|\|^2$	$f(A,B) > (c^A + c^B)^2$ $f(A,B) < (c^A + c^B + m)^2$
A `InFrontOf` B	$\frac{\bar{d}(A,B)\cdot\bar{f}^B(t)}{\|\|\bar{d}(A,B)\|\|\times\|\|\bar{f}^B(t)\|\|}$	$0.9 < f(A,B) < 1.0$
A `ToRightOf` B	$\frac{\bar{d}(A,B)\cdot\bar{r}^B(t)}{\|\|\bar{d}(A,B)\|\|\times\|\|\bar{r}^B(t)\|\|}$	$0.9 < f(A,B) < 1.0$
A `OnTopOf` B	$\frac{\bar{d}(A,B)\cdot\bar{u}^B(t)}{\|\|\bar{d}(A,B)\|\|\times\|\|\bar{u}^B(t)\|\|}$	$0.9 < f(A,B) < 1.0$

Table 1: Example trajectory constraints for relation types.

$\bar{r}(t)$. We define c as the radius of a circumscribing sphere of the entity's bounding box.

A constraint such as in Figure 8 is defined between two entities. We support the constraints listed in Table 1, where for convenience we define the following functions between two entities, A and B, with trajectories $R^A(t)$ and $R^B(t)$ respectively:

- Difference vector:

$$\bar{d}(A,B,t) = (R^A_x(t) - R^B_x(t), R^A_y(t) - R^B_y(t), R^A_z(t) - R^B_z(t))$$

- Norm:

$$||\bar{v}|| = ||(v_1, v_2, v_3)|| = \sqrt{v_1^2 + v_2^2 + v_3^2}$$

The `InsideOf` and `AdjacentTo` constraints in Table 1 are distance functions, ensuring that the trajectories of the two entities are withing a certain radius of each other. In order to achieve `InFrontOf`, `ToRightOf` and `OnTopOf` constraints we constraint of the dot product between the subject entity and the direction vector of the object entity such that the two vectors are at right angles. Note that for the `AdjacentTo` constraint, some value must be chosen for m which indicates a maximum distance allowed between two entities in order to classify them as adjacent. Different types of relations are combined to create the required results. For instance where an entity is to appear in front of another the `InFrontOf` as well as the `AdjacentTo` relation must be applied to ensure that the two entities are near to each other.

Each constraint is specified to hold over a specific time span. Assume that a constraint is specified to hold over time interval $T = [t_0, t_1]$. This means that the constraint, defined with trajectories as functions of t, must hold $\forall t \in T$. We refer to the intervals of time, T, over which a constraint must be satisfied as *universally quantified* variables.

Once constraints are specified for the trajectories of each entity, values must be found for the variables defining the trajectories such that all the constraints are satisfied. As is evident in Table 1 we do not assume linearity over the constraints, and maintain the freedom to choose trajectories of arbitrary degree. The constraint solving process must apply to sets of non-linear constraints and ensure that the constraints are satisfied over the entire specified time intervals.

7.2 Constraint Solving

The lack of linearity in the constraints, and the presence of universally quantified variables means that traditional constraint solving techniques such as linear programming are not applicable. We describe three different constraint solving techniques, each of which has its own merits.

7.2.1 Random walk with time-sampling

Algorithm 1 presents the random walk algorithm for constraint minimisation. Initially each variable in the set of constraints is assigned a random value and the set evaluated using Algorithm 2. A variable is chosen at random from the set of variables and modified slightly. The constraint set is re-evaluated, and if the score improves the new value is kept. The process repeats until a no improvement is made after a specified number of steps, or until manually stopped.

The evaluation procedure takes n samples from the time interval for a constraint, and evaluates a score for the constraint at that point. This score is calculated by evaluating the appropriate function from Table 1, and calculating the difference between the result and the limits indicated in the third column of the table. Negative differences indicate that the constraint is totally satisfied. Positive differences are scaled with a heuristically defined weighting factor that normalises scores across different types of constraints, and gives precedence to certain types of constraints specified by the user.

This approach suffers in number of respects. The sampling approach to evaluation is not completely accurate, in that portions of the trajectories that violate constraints may fall entirely between samples. Also this optimisation approach is very slow at producing satisfactory results. This is because the traversal of the search space cannot be directed in any intelligent manner.

The advantages of the strategies, other than the obvious simplicity, is that it can easily be time bounded in its search. The fact that it always has a current best solution available means that this can be used when a solution is required, even if this solution is not optimal, or fails to satisfy all constraints. In many respects it is similar to other random number based optimization strategies such as genetic algorithms or simulated annealing.

7.2.2 Static objects using constraint dependencies

We find many instances of static objects that are only ever affected by one constraint. For instance, our constraint generation procedure creates a set of constraints that specify which Areas must be adjacent. This is done by examining sequences of `Inside` constraints

Algorithm 1 Random walk optimisation, where C is the set of constraints, V the set of variables, and n the number of samples to take across each time interval.

```
minimise (in: C, V, n)
  v ←random value ∀v ∈ V
  score_V ← evaluate(C,V,n)
  while true
    v_i ← any variable from V
    Δv_i ← random value in range of v_i
    v ← v+Δv
    score_+ ← evaluate(C,V|v_i ← v_i+Δv_i,n)
    score_− ← evaluate(C,V|v_i ← v_i−Δv_i,n)
    if score_+ < score_V
      v ← v+Δv
      score_V ← score_+
    if score_− < score_V
      v ← v−Δv
      score_V ← score_−
```

Algorithm 2 Sampling-based evaluation procedure, where C is the set of constraints, V the set of variables, and n the number of samples to take across each time interval.

```
evaluate (in: C, V, n; out: score)
  foreach c ∈ C
    T ← time interval of c
    t ← T_lower
    score ← 0
    for i← 0 to n−1
      t ← t+width(T)/n
      score ← score+ evaluateScore(c,t)
  return score
```

applying to each Avatar. If an Avatar moves from inside one Area, to inside another Area, then an implicit constraint is created specifying that the two areas must be adjacent. Typically the relations between areas are never specified in a story, but are true for the entire duration of the story. In these cases the time aspect may be removed from the solving process.

A minimisation technique is possible in this context and works as follows: An object A is selected at random, and the set of constraints applying to this object are minimised using Algorithm 1 (with $n = 1$). All objects involved in this process are added to a set S, membership of which implies that no further modifications may be made to locations of objects in S. The constraints applying to each object in S and any object not already in S are then minimised.

The problem with this approach, besides the fact that it is limited to non-temporal constraints, is that if a cycle of constraints exists between set of objects, then at some stage this cycle is broken, which means that there will always be unsatisfied constraints, as is the case between the `dressing-room` and `outside the door`, constraint 6 in Figure 9. Since both of these areas are already in the set S, this constraint cannot be minimised and so the cycle of constraints is broken.

This process is attractive since it results in tight placement of areas, as shown in Figure 9. In addition, a solution is achieved within a small finite period of time.

7.2.3 Interval-based constraint solving

Interval arithmetic is a field of mathematics that define operations on closed intervals of numbers. Let $\mathbb{R}$ be the set of real numbers.

```
1.  outside the door ADJACENT_TO  stage
2.  stage            ADJACENT_TO  dressing-room
3.  dressing room    ADJACENT_TO  foyer
4.  cellars          ADJACENT_TO  stage
5.  room             ADJACENT_TO  staircase
6.  dressing-room    ADJACENT_TO  outside the door
```

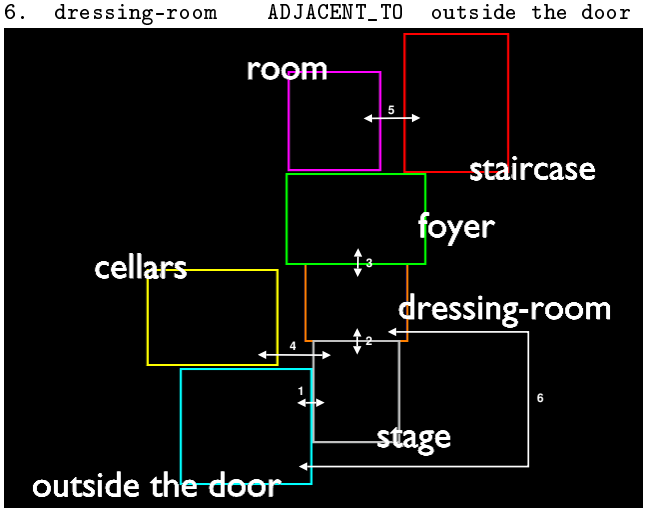


Figure 9: Example solution for a set of static constraints, using a dependency-based approach.

The set of numbers that can be represented on a computer is $\mathbb{F} \subset \mathbb{R}$. A *floating point interval* is a set of real numbers bound on either side by floating point numbers. Formally, given $g \in \mathbb{F}$ and $h \in \mathbb{F}$, then $[g,h] = \{g \leq r \leq h | r \varepsilon \mathbb{R}\}$ [Moore 1966]. Therefore, the interval $[g,h]$ contains every real number between (and including) g and h. An interval is denoted using uppercase (for example, $I = [g,h]$).

Let $A = [a,b]$ and $B = [c,d]$. Various interval operators are defined as *interval extensions* to their real-valued counterparts [Moore 1966]:

- Addition: $A \oplus B = [a+c, b+d]$
- Subtraction: $A \ominus B = [a-d, b-c]$
- Multiplication: $A \otimes B = [min(S), max(S)]$ where $S = \{a*c, a*d, b*c, b*d\}$
- Division: $A \oslash B = A \otimes 1/B$ where $0 \notin B$, undefined otherwise.

Each of the above operators can be used in place of their real-valued counterparts in order to transform expressions such as those in Table 1 into their *natural interval extensions*. Constraint solving techniques exist for interval arithmetic, specifically designed for solving over universally quantified variables [Benhamou et al. 2004; Benhamou et al. 1999]. We have implemented these techniques (described in more detail in previous work [Glass and Bangay 2007]) and have applied them to a limited set of constraints over simple parabolic trajectories.

We find this method works well for our time-based constraints, but find that the location of a solution is often a very lengthy process, especially as further variables are added. We are currently investigating methods for interval-based constraint *optimisation* rather than solving, which allows for the relaxation of constraints, and possible solutions in shorter times. In this manner we will combine traditional search methods with the ability to work with intervals of numbers, as well as exploit the powerful pruning techniques that have been developed for interval constraint solving.

7.3 Populating virtual environments

Given trajectories for entities that reflect the requirements of the story, the next task in creating a virtual environment is selecting bodies for each entity, assigning trajectories, and directing a camera to film the correct portions of the 3D scene.

Rendering entities requires suitable object models; geometry that appropriately represents the shape of the entity. Information regarding requirements for a representative object model can be sparse and in some cases only consists of a few words. We have a library of pre-built objects, and have devised means for annotating it so that it is suitable for matching against textual descriptions. The library contains models for which we have defined default semantic properties. In particular each model in our library is standardised to be of unit height, and is assigned a scaling factor which, when applied to the object, transforms it to its typical size in the real world. This scaling factor makes provision for modifications to the object's size based on information derived from, for example, adjectives that specify the entity's size.

In addition to size, each object is by default centered at the origin, facing down the positive z-axis, with the upward direction parallel to the y-axis, and the rightward direction parallel to the x-axis. The object is transformed such that these directions align with the $\bar{f}$, $\bar{u}$ and $\bar{r}$ directional vectors, described in Section 7.1.

Most importantly, each object in our database is assigned a set of keywords that describe it, specified manually. Synonyms of the provided keywords are also included, and the annotator has the option of associating appropriate hypernyms (more general terms), or hyponyms (more specific terms), to the object (provided using WordNet [Fellbaum 1998]). These keywords are compared with the search term when searching for a desired model.

We wish to find models for the types of entities we define earlier: namely Avatar, Object and Area. Each of these entity types contain instructions and parameters that define what category of model is appropriate for its specific type.

Representing Avatars Avatars (unless explicitly stated in the the text) are assumed to be human, and so only humanoid models are appropriate for this category. The Avatar entity contains a gender parameter which is derived using indicators in the text, and this parameter further refines the type of models appropriate: models annotated with either male or female keywords.

Representing Objects Objects do not have any specific parameters, making the search for appropriate models more difficult. The triggers associated with an Object are used as search terms when querying the model database. However, it is unreasonable to expect a library of object models to contain a model for every object in the universe. As such we implement a method of search term generalisation to locate models for all entities. If no matching keyword exists in the object library for a specific search term, then all the synonyms of the term returned using WordNet [Fellbaum 1998] are tried. If no match exists in this case then the immediate hypernym of the term is used as a search term, and this process is repeated. For example, our library contains no object model with the keyword "spaniel". However, a hypernym of this noun is "dog", of which our database contains numerous models. The process of generalisation continues until the term can no longer be generalised, and a default placeholder object is selected. Procedural generation of objects has also been investigated for this purpose [Morkel and Bangay 2006].

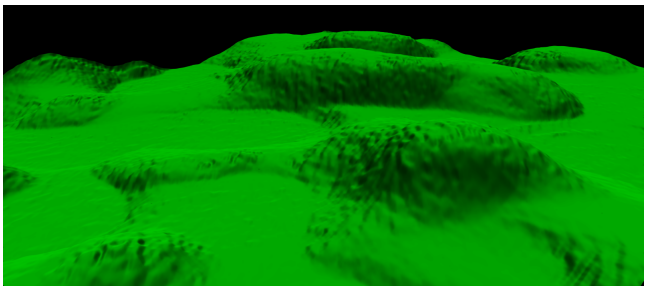


Figure 10: Procedurally generated terrain

Representing Areas Areas are an interesting case in that they cannot be represented using a single object model. Most Areas refer to rooms in a building, which are easily represented using large cubes. However, other general areas are often referenced, such as "the city", or "valley" (from *The Famous Five: Five on a Treasure Island* by Enid Blyton), in which case a cube-shaped area does not suffice. In such instances images such as discussed in Section 6 may be used as backdrops to the scene. Alternatively we are investigating procedural techniques for the automatic generation of terrain and cities [Glass et al. 2006]. The terrain is created by sampling multiple frequencies of Fractional Brownian Motion and eroding it via hydraulic erosion [Musgrave et al. 1989]. Figure 10 shows an example of a rolling hills terrain. A similar process of search term generalisation will be used until a term is recognised that invokes an available terrain generation procedure.

7.4 Creating video from virtual environments

During the constraint creation phase we also extract constraints for an abstract camera object. Each time a trigger that points to an entity is encountered in the text, a constraint is created that ensures the camera is located in front of that particular entity at the correct time in the scene. Hence the trajectory of the camera is solved in conjunction with the rest of the entities on the 3D scene.

Once trajectories of objects have been defined, all that remains is to assemble a rendering of the fiction text. In particular we wish the graphics that are displayed to correspond to the text which is being played via audio and displayed in the subtitles. Each segment of text in the presentation time-line requires a corresponding 3D rendering, and this is found by filming the appropriate portion of the 3D scene. This portion is rendered over the corresponding scene-time, as derived using the relation illustrated in Figure 3. Time compression or dilation may be required to fit the scene-time sequence into a presentation time segment of a different duration. Each segment is created by rendering from the point of view of the camera over the specified portion of the scene time-line. An example result of this process is presented in Figure 1, where the avatar representing `ballet-dancers` is first shown to be inside the `dressing-room` according to the corresponding text, while the next frame shows the `ballet-dancers` when they were on the `stage` as described by the "flashback" text.

8 Integration

We use the sequencer facility available in the Blender modeling package (`http://www.blender.org`), which allows for insertion of audio and image files, as well as portions of animated 3D scenes designed within the same package. This means that portions of the

3D scene do not need to be pre-rendered, and only those sections indicated in the sequencer are rendered during compilation.

Figure 11 shows a compilation sequenced in the Blender modeling package. This compilation was generated automatically using the processes described in this paper demonstrating the use of the methods presented in this paper. The figure also indicates mixing channels, used for blending the images and subtitles or 3D scene renderings and subtitles. It is evident in Figure 11 that we have achieved the goal of compiling alternative representations of the input text into a single multimodal sequencing paradigm, the design of which is indicated in Figure 2.

9 Conclusion

This paper successfully demonstrates a range of techniques that generate multimodal versions of fiction text and assemble them into a single, synchronised representation. These techniques automate processes that previously required human intervention. We have translated stories into constraints that allow us to generate equivalent virtual environments. The presentation of text, audio and images correlates to the unfolding story in the virtual environment through the relationship between the presentation and scene time-lines.

Contributions include:

- using audio synthesised from the original text to create a presentation time-line, from which timing and alignment information can be derived.
- using different voice parameters to improve quality of audio presentation.
- generating speech-bubbles and avatars to enhance the comprehension of subtitles.
- constructing a scene time-line for extraction of time-based constraints, and synchronising the events in the virtual world to the presentation media.
- populating a virtual world providing a 3D representation of the text.

Future work will focus on generating better quality virtual environments representing the fiction text; specifically the creation of a constraint solver capable of solving the expressions we define in this paper and in quantities sufficient to handle an entire book. [1]

References

AKERBERG, O., SVENSSON, H., SCHULZ, B., AND NUGUES, P. 2003. Carsim: An automatic 3D text-to-scene conversion system applied to road accident reports. In *Research Notes and Demonstrations Conference Companion, 10th Conference of the European Chapter of the Association of Computational Linguistics*, Association for Computational Linguistics, Budapest, Hungary, 191–194.

[1]This work was undertaken in the Distributed Multimedia Centre of Excellence at Rhodes University, with financial support from Telkom SA, Business Connexion, Comverse, Verso Technologies, Tellabs and SorTech THRIP, and the National Research Foundation. The financial assistance from the Henderson Scholarship and NRF Scarce Skills Scholarship towards this research is hereby acknowledged.

BACK, M., GOLD, R., AND KIRSCH, D. The SIT book: audio as affective imagery for interactive storybooks. In *CHI '99: CHI '99 extended abstracts on Human factors in computing systems*, ACM Press, New York, NY, USA, 202–203.

BADLER, N. I., BINDIGANAVALE, R., ALLBECK, J., SCHULER, W., ZHAO, L., AND PALMER, M. 2000. *Parameterized action representation for virtual human agents*. Embodied conversational agents. MIT Press, ch. 9, 256–284.

BENHAMOU, F., GOUALARD, F., GRANVILLIERS, L., AND PUGET, J.-F. 1999. Revising hull and box consistency. In *Proceedings of the sixteenth International Conference on Logic Programming (ICLP'99)*, MIT Press, Las Cruces, New Mexico, United States, 230–244.

BENHAMOU, F., GOUALARD, F., LANGUÉNOU, É., AND CHRISTIE, M. 2004. Interval constraint solving for camera control and motion planning. *ACM Transactions on Computational Logic (TOCL) 5*, 4, 732–767.

BILLINGHURST, M., KATO, H., AND POUPYREV, I. 2001. The magicbook - moving seamlessly between reality and virtuality. *Computer Graphics and Applications 21(3)* (May-June), 2–4.

BINDIGANAVALE, R., SCHULER, W., ALLBECK, J. M., BADLER, N. I., JOSHI, A. K., AND PALMER, M. 2000. Dynamically altering agent behaviors using natural language instructions. In *AGENTS '00: Proceedings of the fourth international conference on Autonomous agents*, ACM Press, New York, NY, USA, 293–300.

CHU, Y.-C., WITTEN, I. H., LOBB, R., AND BAINBRIDGE, D. 2003. How to turn the page. In *JCDL '03: Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital libraries*, IEEE Computer Society, Houston, Texas, 186–188.

CLAY, S. R., AND WILHELMS, J. 1996. Put: Language-based interactive manipulation of objects. *IEEE Computer Graphics and Applications 16*, 2 (March), 31–39.

COYNE, B., AND SPROAT, R. 2001. Wordseye: an automatic text-to-scene conversion system. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, ACM Press, New York, NY, USA, 487–496.

FELLBAUM, C., Ed. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA, USA.

GLASS, K., AND BANGAY, S. 2005. Evaluating parts-of-speech taggers for use in a text-to-scene conversion system. In *SAICSIT '05: Proceedings of the 2005 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries*, South African Institute for Computer Scientists and Information Technologists, Republic of South Africa, 20–28.

GLASS, K., AND BANGAY, S. 2006. Hierarchical rule generalisation for speaker identification in fiction books. In *SAICSIT '06: Proceedings of the 2006 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing couuntries*, South African Institute for Computer Scientists and Information Technologists, Republic of South Africa, 31–40.

GLASS, K., AND BANGAY, S. 2007. Constraint-based conversion of fiction text to a time-based graphical representation. In *SAICSIT '07: 2007 annual research conference of the South African institute of computer scientists and information technologists*, South African Institute for Computer Scientists and Information Technologists, Republic of South Africa.

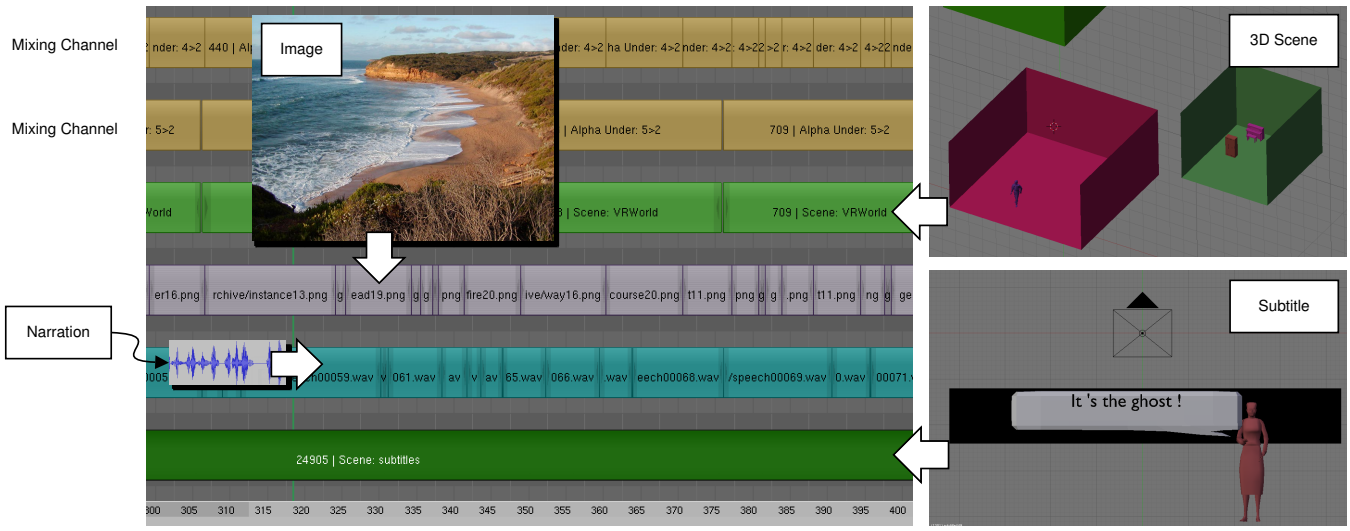


Figure 11: Example sequencer containing automatically generated segments from different representations of the fiction text.

GLASS, K. R., MORKEL, C., AND BANGAY, S. D. 2006. Duplicating road patterns in South African informal settlements using procedural techniques. In *Afrigraph '06: Proceedings of the 4th international conference on Computer graphics, virtual reality, visualisation and interaction in Africa*, ACM Press, New York, NY, USA, 161–169.

HOOD, M. 2004. Creating a voice for festival speech synthesis system. Tech. rep., Computer Science Department, Rhodes University, Grahamstown, South Africa, November.

JOSHI, D., WANG, J. Z., AND LI, J. 2004. The story picturing engine: finding elite images to illustrate a story using mutual reinforcement. In *Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval*, ACM Press, New York, NY, USA, 119–126.

LU, R., AND ZHANG, S. 2002. *Automatic Generation of Computer Animation: using AI for movie animation*, vol. 2160 of *Lecture Notes in Computer Science*. Springer, Berlin.

MA, M. E. 2002. CONFUCIUS: An intelligent multimedia storytelling interpretation and presentation system. Tech. rep., School of Computing and Intelligent Systems, University of Ulster, Magee, September.

MOORE, R. E. 1966. *Interval Analysis*. Prentice-Hall, Inc., New Jersey, USA.

MORKEL, C., AND BANGAY, S. 2006. Procedural modeling facilities for hierarchical object generation. In *Afrigraph '06: Proceedings of the 4th international conference on Computer graphics, virtual reality, visualisation and interaction in Africa*, ACM Press, New York, NY, USA, 145–154.

MUSGRAVE, F. K., KOLB, C. E., AND MACE, R. S. 1989. The synthesis and rendering of eroded fractal terrains. In *Proceedings of the 16th annual conference on Computer graphics and interactive techniques*, ACM Press, 41–50.

PIESK, J., AND TROGEMANN, G. 1997. Animated interactive fiction: Storytelling by a conversational virtual actor. In *VSMM '97: Proceedings of the 1997 International Conference on Virtual Systems and MultiMedia*, IEEE Computer Society, Washington, DC, USA, 100.

TABORDET, F., PIED, F., AND NUGUES, P. 1999. Scene visualization and animation from texts in a virtual environment. *Journal for the integrated study of artificial intelligence, cognitive science and applied epistemology 15*, 4, 339–349.

TAPANAINEN, P., AND JÄRVINEN, T. 1997. A non-projective dependency parser. In *Proceedings of the 5th Conference on Applied Natural Language Processing*, Morgan Kaufmann Publishers Inc., Washington, DC, USA, Association for Computational Linguistics, 64–71.

ZENG, X., MEHDI, Q. H., AND GOUGH, N. E. 2003. Shape of the story: Story visualization techniques. *Proceedings of the Seventh International Conference on Information Visualization (IV'03)*, 144–149.

ZENG, X., MEHDI, Q. H., AND GOUGH, N. E. 2005. From visual semantic parameterization to graphic visualization. In *IV '05: Proceedings of the Ninth International Conference on Information Visualisation (IV'05)*, IEEE Computer Society, Washington, DC, USA, 488–493.

ZHANG, J., BLACK, A., AND SPROAT, R. 2003. Identifying speakers in children's stories for speech synthesis. In *Proceedings of EUROSPEECH 2003*, Institute for Perceptual Artificial Intelligence, Geneva, Switzerland.

Image Sources

Image sources for Figure 1 are as follows (in order from left to right):

- `http://www.wm.edu/theatre/DressingRoom.jpg`
- `http://cucinatestarossa.blogs.com/photos/uncategorized/milk_cream_eggs.jpg`
- `http://the-falcon1.tripod.com/sitebuildercontent/sitebuilderpictures/.pond/snow_cover-ithaca-ballet-and-modern-dance-cities.jpg.w300h404.jpg`
- `http://www.theocracyofthepale.com/images/ghost.jpg`
- `http://www.holycross.net/images/Easter%25202002/Front%2520Door.jpg`