Entertainment Computing 5 (2014) 33–41

Contents lists available at SciVerse ScienceDirect

Entertainment Computing

journal homepage: ees.elsevier.com/entcom

Draw your own story: Paper and pencil interactive storytelling ☆

Edirlei Soares de Lima [a,*], Bruno Feijó [a], Simone D.J. Barbosa [a], Antonio L. Furtado [a], Angelo E.M. Ciarlini [b], Cesar T. Pozzer [c]

[a] Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, Rio de Janeiro, Brazil
[b] Universidade Federal do Estado do Rio de Janeiro, Departamento de Informática Aplicada, Rio de Janeiro, Brazil
[c] Universidade Federal de Santa Maria, Departamento de Eletrônica e Computação, Santa Maria, Brazil

ARTICLE INFO

Article history:
Received 22 September 2012
Revised 21 May 2013
Accepted 30 June 2013
Available online 9 July 2013

Keywords:
Interactive storytelling
Augmented reality
Sketch-based interface

ABSTRACT

Drawing is a primary human skill that has been used for thousands of years as a visual complement to written and oral storytelling. The advent of interactive narratives brings the possibility of interaction to the traditional stories. In this paper, we present a storytelling system able to dramatize interactive narratives in augmented reality over a conventional sheet of paper. The system allows users to freely interact with virtual characters by sketching objects on the paper. Users interacting with the system can thus indirectly affect the characters' decisions, even to the point of radically subverting the storyline. We validate the proposed system with a user study conducted with 21 participants. The results show that the use of hand drawings as a form of interaction improves user satisfaction and experience and the system usability.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

Humans have chosen to express themselves through hand drawings since at least the prehistoric Paleolithic period. We may well imagine our prehistoric predecessors gathering inside caves to both listen to and "read" stories about their hunting feats. Nowadays, we tell stories through a variety of media, from magazines and books to movies and video games. One of the most radical changes brought about by digital storytelling, however, goes beyond the media in which the story is conveyed. "Readers" (henceforth called users) have now the possibility to interact with the story, assuming an increasingly active role in its unfolding. Some of the challenges we currently face concern the mechanisms we provide users to interact with the story line. We need to design ways for users to engage in the story, without limiting their sense of immersion in the fictional world.

HCI researchers have focused on immersive user interfaces over the last fifteen years, from different viewpoints: multimodal interfaces [19,20], virtual reality [19], and augmented reality (AR) [17,18]. However, few immersive systems are devoted to interactive storytelling, and even then they require special devices, such as CAVE-like immersive displays [22] and see-through head-worn displays [21]. Complex device installations make the immersion experience less natural and creative. We believe that simple sketch-based interactions may have a positive impact on the way users interact with digital stories.

Sketch-based interaction has been used in engineering, education, and 3D modeling [23,34], and it is a permanent research topic since Ivan Sutherland proposed his famous SketchPad system [24] in the sixties. Those systems use special input devices (such as tablets) or projection displays. Sketch-based interactive narrative systems have been proposed [2], but the focus is not on the narrative structure. AR systems using sketch-based interaction and ordinary paper and pencil have also been proposed [25,1]. However, as far as we are aware, no researchers have reported the development of AR systems for interactive storytelling that use paper and pencil as an interaction media.

We believe that hand drawing in interactive storytelling can be used to stimulate user participation, increase the sense of authoring, and provide more immersive story worlds. In this paper, we present an interactive storytelling system that combines an augmented reality (AR) interface with a sketch-based interaction interface using a conventional sheet of paper. We propose a dramatization layer that can be added to the top of any plot generation model that produces events in the form of predicates. This layer is composed by an emotional and social model, an action planner, and a sketch-based interface that uses everyday paper and pencil. Users can freely interact with the virtual characters by sketching objects on the paper, which are recognized by the system and converted into objects in the 3D story world. Furthermore, we present a user evaluation study that shows how this new interface impacts the usability and overall user experience.

☆ This paper has been recommended for acceptance by Junia Anacleto, PhD on Physics.

* Corresponding author. Tel.: +55 21 3527 1510; fax: +55 21 3527 1530.
E-mail address: elima@inf.puc-rio.br (E.S. de Lima).

1875-9521/$ - see front matter © 2014 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.entcom.2013.06.004

We implement the dramatization layer on the top of our previous plot generation system [27–29] that uses a nondeterministic planning algorithm. In the present paper, although the nondeterminism aspects of our system are an important question, we focus on the dramatization phase only. In this phase, we dramatize the events through actions performed by emotional and social characters that plan their actions in an environment being changed by users sketching objects on a piece of paper. The paper is organized as follows. Section 2 describes related works. Section 3 presents the architecture and implementation of the paper-and-pencil interactive storytelling system. Section 4 describes a simple application and a user evaluation study. Section 5 contains the concluding remarks.

2. Related work

The present work involves the combination of hand-drawn sketches with a mixed reality interactive storytelling system. This section describes some previous works that have also applied these ideas in the context of interactive narratives and games.

Vogelsang and Signer [2] and Kuka et al. [3] propose to use hand-drawn sketches to interact with digital narratives and to create virtual objects, respectively. However, their systems depend on special pen and paper[1] to record and send the user's strokes to a computer. Bayon et al. [4] combine a mixed reality and collaborative storytelling environment in which children use a variety of devices (PDAs, scanners, bar codes and a large screen display) to tell stories. Jhala et al. [33] present a system that uses a pen-based interface for authoring scenarios and sketching 2D storyboard frames that are rendered in 3D on a game engine. However, this system is an authoring tool for creating machinima (which is an art form that uses 3D game engines to produce films), rather than an interactive storytelling system.

The use of mixed reality environments in the context of interactive narratives has also been the focus of some research projects. Dow et al. [5] present an AR version of the desktop-based interactive drama Façade [6]. With a similar approach, Cavazza et al. [7] present an interactive storytelling application that captures the user's video image and inserts him/her in a world populated by virtual actors. Users are able to interact with the virtual actors using body gestures and natural language speech. While Dow et al. [5] bring the virtual characters to the real world, Cavazza et al. [7] place the user inside the virtual world. Zhou et al. [16] explore the use of tangible cubes as interaction interface for mixed reality interactive storytelling. In their system, the storytelling process is controlled by two cubes: the first cube is used to navigate through different scenes of the story, whilst the second is used to choose different items needed in the story. The users visualize the stories in AR over the cubes.

In the AR gaming context, hand-drawn sketches have been used by Hagbi et al. [1] as a content-authoring tool for drawing scenarios on a sheet of paper, whereas Huynh et al. [8] use physical tokens to represent individual towers on a board game. Leitner et al. [9] explore the interaction between real world objects and virtual objects, also in board games.

None of the aforementioned works appears to combine all the characteristics of the system we proposed here, that is: a paper and pencil interactive storytelling tool with a sketch-based AR interface that allows an easy and more natural way of influencing the ongoing story. The use of computer vision techniques to recognize the user sketches, in lieu of special equipment (*e.g.*, digital pens, special papers, tablets) makes the application more flexible and readily available to the general public. The mixed reality visualization interface encourages user interaction and favors deeper immersion in the story world. Moreover, the use of hand-drawn sketches instigates curiosity and involvement, which can help interactive narratives to become a potential educational tool in children's environments.

3. Paper and pencil interactive storytelling

Our paper and pencil interactive storytelling system combines a sketch-based interface with an AR visualization interface. It is composed of a computer equipped with a conventional webcam, an ordinary sheet of paper with a fiducial marker printed on it, and a common pencil (Fig. 1). Furthermore, the webcam can be replaced by virtual reality glasses to create even more immersive experiences, if available.

In the system, stories are graphically represented in AR over the paper, which creates the illusion that the sheet of paper is a virtual world populated by virtual characters. The entire world may comprise several sheets of paper, each one representing a different location in the virtual world. Users can switch between places by changing the paper shown to the camera or by pointing the camera to other sheets of paper. Both users and system can distinguish the places based on the fiducial markers on each page.

Users act as gods of the virtual world, in a way reminiscent of the *deus ex machina* of classical theater. While interacting with the system, users can influence the decisions made by the virtual characters by sketching objects on the paper, which are transferred to the virtual world. For example, a hero may not have enough strength to slay the villain with his bare hands, but if the user draws a sword close to the hero's position in the paper, the sword will be transferred to the virtual world and taken by the hero, who will now be able to defeat the villain.

In the AR environment, users have the freedom to move the camera and watch the scenes from different angles. Moreover, like film directors, they have the power to change the perspective of the stories simply by choosing to focus on a different virtual place, which generates different events and thus changes the story unfolding.

The paper and pencil interactive storytelling system is composed of three main modules: the character story planner, the sketch recognition interface, and the AR dramatization system (Fig. 2). The story planner handles the actions of several virtual autonomous characters, each one introduced with predefined goals, whose behavior may however be redirected via user interactions. The sketch recognition system consists of a support vector

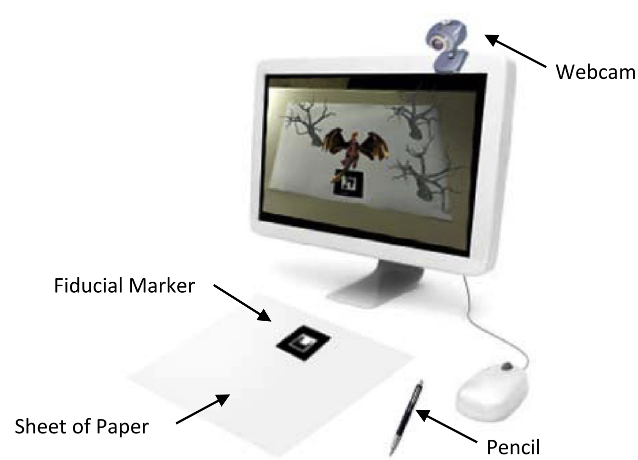


Fig. 1. The simple environment of the proposed system.

[1] Anoto Digital Pen – http://www.anoto.com.

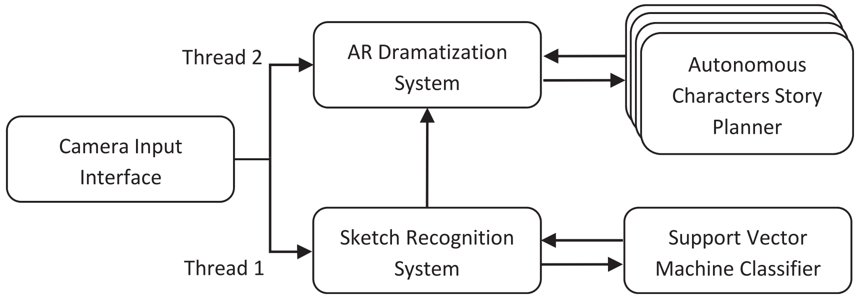


Fig. 2. Parallel system architecture.

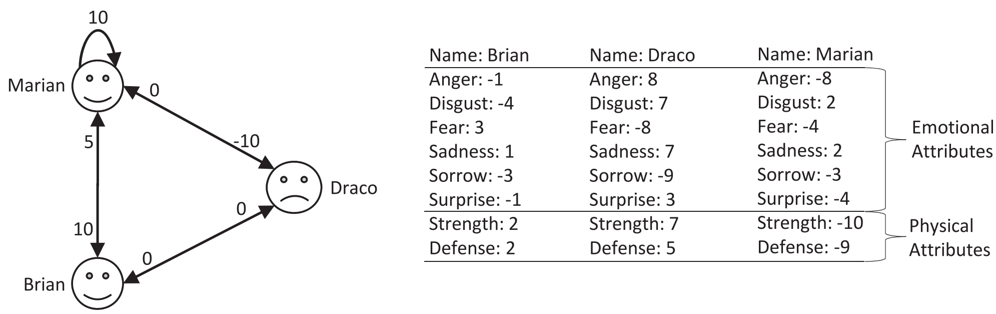


Fig. 3. The proposed multi-character network.

machine (SVM) classifier trained to recognize a set of sketches users draw on a sheet of paper, which are then captured by the camera. The AR dramatization system controls and renders the virtual world superimposed over the real world objects, creating a mixed reality environment. If a fiducial marker is found on the image, the system renders the virtual world objects and characters according to the virtual location identified by the marker. The AR dramatization system uses the ARToolKit Library.[2]

The parallel architecture of the system is important to guarantee that there will be no noticeable delays in the rendering process – which is currently limited to 30 frames per second, due to the camera capture speed. Since the recognition of user sketches is the most expensive process in the system, it must be executed in a separate thread, so that the system is able to efficiently render the output images in real-time.

3.1. Story planner

Interactive storytelling systems can follow three basic approaches: plot-based [27], character-based [30], or a hybrid approach [26]. In this paper, we add a layer on the top of a nondeterministic hybrid approach to interactive storytelling found in some of our previous works [27–29]. A nondeterministic planning algorithm generates a plan in which events allow characters to try achieving goals without necessarily succeeding [29]. Furthermore, events can be specified by nondeterministic automata [28], in which the arcs represent short episodes that we call ‘‘actions’’. In this paper, we propose an interactive layer that can represent an action in those automata. As the actions are nondeterministic, the interactions that occur via this layer can influence the rest of the story. The story evolves towards surprising outcomes depending on emotional and physical states that characters can attain as a result of the hand drawn user interventions. The example implemented in the prototype corresponds to a short story within a swords and dragons genre. In this story, a villain (dragon) kidnaps a narcissistic princess, who can easily get depressed, and a brave knight tries to rescue her.

The emotional, physical, and social attributes of the characters are modeled as a multi-character network (Fig. 3), in which nodes represent characters and bidirectional arcs define affection relationships in the social environment of the story. Each node has the name of the character and the values of the emotional/physical attributes. Affections are not reciprocal, that is affection(i,j) is not necessarily equal to affection(j,i), except when there is a self-affection situation. Affection values vary within the interval $[-10,10]$.

The emotional model adopted by our new planner uses the six emotions proposed by Ekman and Friesen [10], but we consider them lying on six emotion axis with negative and positive sides that represent opposite emotions: [calmness, anger], [liking, disgust], [confidence, fear], [joy, sadness], [cheeriness, sorrow], and [anticipation, surprise]. The values in each axis are numbers within the interval $[-10,10]$. In this model, sorrow is not a synonym of sadness, but a sense of loss or a sense of guilt and remorse. For the sake of simplicity, we refer to an axis by the name of its positive side. The sign ($-$ or +) does not mean destructive or constructive emotions, but a connotation of drama impact and opposite states. Turning points or dramatic situations are more easily described with the aforementioned six positive emotions, like anger and disgust. In this model, we should be careful when referring to high levels of opposite emotions. For instance, very high levels of joy (*i.e.* very low levels of sadness) correspond to negative numbers on the axis $[-10,10]$.

In the proposed model, emotions can be combined to form a new emotion, for instance: love = joy + liking + confidence. Also, we can refer to extreme values on an axis as being special emotions, *e.g.*: grief = very high levels of sadness and ecstasy = very high levels of joy (that is, very low levels of sadness).

The author of the story is responsible for classifying the characters within some personality stereotypes consistent with their attributes, according to which the logical rules that determine the characters’ behaviors should be formulated. For example, ‘‘good people’’ will always try to help and defend people in danger, especially their beloved ones; ‘‘bad people’’ will always try to hurt people that they hate, and to force everyone to do what they want.

[2] http://www.hitl.washington.edu/artoolkit/.

The story planner used by the paper-and-pencil interactive layer is defined by general rules and production rules. General rules express general knowledge about the genre, such as "if a person is not at home he/she will be unprotected". For example: $\forall X$ *currentPlace(X)* $\neq$ *home*$(X) \rightarrow \sim$*protected*(X), where $\sim$ denotes negation.

Production rules concern actions and have the following form:

$$\text{CONDITIONS} \rightarrow \text{actions}(\text{CONCRETE\_ACTIONS}, \text{ATTRIBUTE\_CHANGES})$$

where CONDITIONS is a conjunction of observed facts; CONCRETE_ACTIONS is a list of concrete actions, such as go, take, hit, and kidnap; and ATTRIBUTE_CHANGES is a list of increments and decrements to the indicated characters' attributes using the operators *add(X,Attribute,Increment)* or *addr(X,Y,Attribute,Increment)*, *e.g. add(marian,sadness,9)* increments Marian's current value of sadness by 9 and *addr(marian,brian,affection,5)* increments Marian's affection to Brian by 5. In the proposed model, we work with attribute operators that return the current level of a specific attribute for a specific character *X*. These operators use the same terminology of the multi-character network, *e.g. sadness*(*X*) and *affection*(*X*, *Y*). Also the model has an important operator that confirms if a specific *Object* drawn by the user can be seen by a character *X*: *cansee*(*X*,*Object*). For example, if the user draws a mirror on the paper close to princess Marian, *cansee*(*marian*,*mirror*) will be true. Users interacting with the system can indirectly affect the characters' decisions, even to the point of radically subverting the storyline. The prototype application includes 42 production rules. The following sentences are examples:

(a). Bored princesses always become sad:

$$\forall X princess(X) \wedge protected(X) \wedge surprise(X) < -2$$
$$\rightarrow actions([], [add(X, sadness, 9)])$$

(b). Bored and sad princesses always start searching for new emotions:

$$\forall X princess(X) \wedge protected(X) \wedge surprise(X)$$
$$< -1 \wedge sadness(X) > 6$$
$$\rightarrow actions([go(X, church)], [])$$

(c). Drawing objects on paper (detected by *cansee*) causes changes in attribute levels:

$$\forall X princess(X) \wedge free(X) \wedge cansee(X, mirror)$$
$$\rightarrow actions([go(X, mirror), take(X, mirror)],$$
$$[add(X, surprise, 3), add(X, sadness, -3)])$$

(d). Actions can be more complex than simple movements, such as the one that defines the act of kidnapping:

$$\forall X \forall Y villain(X) \wedge affection(X, Y) < -8 \wedge \sim protected(Y)$$
$$\wedge free(Y) \rightarrow actions([go(X, currentPlace(Y)), hit(X, Y),$$
$$(X, Y), go(X, home(X))], [])$$

General knowledge and production rules may produce unexpected but coherent behaviors as a logical consequence from the rules. For instance, a depressed princess left alone with a bottle of poison may kill herself if intense unhappy events occur.

The basic layers of our previous interactive storytelling system use Constraint Logic Programming to specify rules used in plot generation [27,29] and dramatization control [28]. In the present work, the dramatization control uses a C++ forward chaining procedure instead of logic programming procedures because performance is a critical issue in a sketch-based interactive storytelling system. We implement a story loop with a fixed time step (15 s) to update the characters' states, and in every loop all the rules are re-executed. When the user interferes with the story, the states are updated as soon as a new object is drawn on the sheet of paper.

3.2. Recognition of hand-drawn sketches

The recognition of hand-drawn sketches is an active field of research in computer vision. Many papers have already addressed this problem and several approaches have been investigated [23,25,1]. However, few of them address the recognition of freehand sketches in real-time using a conventional camera as the system input; furthermore, none has tackled storytelling issues. The interactive storytelling system presented in this paper aims to explore the visualization freedom provided by AR environments. The system recognizes freehand sketches in real-time and deals with different camera angles and lighting conditions.

The process of recognizing hand-drawn sketches can be divided in two phases: (i) a pre-processing phase, in which the background of the image frame (captured by the camera) is removed through several segmentation steps; and (ii) a recognition phase, in which the drawings are identified in the segmented images and classified according to a set of previously specified objects.

3.2.1. Pre-processing phase

The pre-processing phase aims to eliminate the background of the input image and highlight the drawing forms. Assuming that drawings are always produced on a blank paper and the strokes contrast with the paper background, the sketches can be clearly distinguished by their contours. However, the image to be processed does not always consist of the sketches and the blank paper alone, since there may exist any number of background objects, which must be found and then ignored by the system in order to reduce the number of candidate drawings.

The approach used in this work to segment the image and remove the background is based on the application of five increasing threshold levels and a canny edge detector over the input image. This process generates six new images containing the candidate drawings. The use of several threshold levels is important to ensure that the drawings may be identified by the system even with different lighting conditions. The threshold levels and the canny edge detector reduce the number of objects on the image, but are still not enough to clear all the background. To completely clear the background, the system uses the paper rectangular border as a boundary to ignore all the objects that are outside it. The sheet of paper is recognized as the largest rectangle on the image. To find it, the system analyzes the contours of all objects at all threshold levels. The paper rectangle will be the one with the largest area having four vertices forming angles of approximately 90 degrees. The AR marker, located inside of the paper, is also removed at the pre-processing phase to avoid being classified as a user drawing. The segmentation process is illustrated in Fig. 4.

After removing the background, all remaining objects on the segmented images are ready to be classified in the recognition phase. Providing a collection of several samples from different segmentation levels to the classifier increases the chances of finding the drawings independently of the stroke width and illumination conditions.

3.2.2. Recognition phase

In the recognition phase, the segmented sketches are classified according to a predefined set of drawings. The correct recognition of a sketch can be seen as a classification problem, wherein the system knows a set of drawings (a vocabulary) and must recognize a new drawing based on its similarity to some member of the known set. To perform this classification, the system uses a support vector machine (SVM) classifier trained with structural features to classify hand-drawn sketches. SVM [11] has proved effective in many

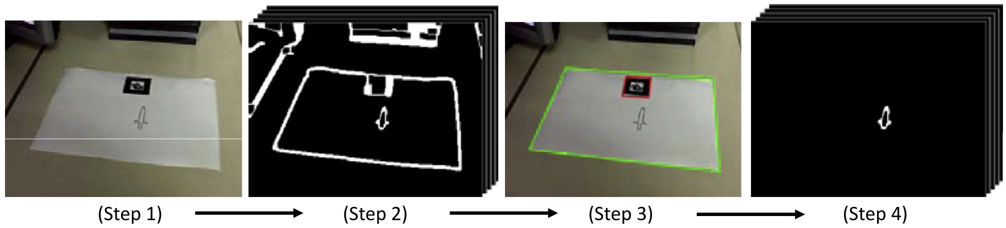


Fig. 4. Segmentation process. (Step 1) input frame captured by the camera; (Step 2) segmented images resulting from the application of increasing threshold levels and the canny edge detector; (Step 3) detection of the paper rectangle and the marker; (Step 4) set of candidate drawings found at the segmentation levels after the elimination of the objects that are outside the paper rectangle and inside the marker square.

real-world applications, such as in systems for detecting microcalcifications in medical images [12], automatic hierarchical document categorization [13], and 3D camera control [14].

The SVM structure is composed of the output classes (the vocabulary of hand-drawn sketches understood by the classifier) and the features vector (numerical features characterizing the hand-drawn sketches). In a training phase, the classifier uses the features to create a pattern that, in the prediction process, classifies unknown input features vectors in one of the output classes. The features vector adopted in the implementation of our classifier is composed of seven moment invariants, which are extracted from the candidate drawings found on the segmented images. The Hu descriptors, proposed by Hu [15], are based on non-orthogonalised central moments that are invariant to image rotation, translation, and scale. This invariance requirement is essential for allowing the classifier to recognize the drawings from different camera positions and angles. The combination of those seven Hu descriptors uniquely characterizes a specific pattern or shape. Computing Hu descriptors is straightforward and fast, and hence this method is widely used in real-time computer vision applications.

Before using the SVM to recognize hand-drawn sketches, the classifier must be trained. The training process consists of capturing (from different angles) several images of hand-drawn sketches, and then processing these images to segment the background and extract the drawing features used by the SVM. The collection of features vectors, extracted from the training images and associated with a specific drawing class, composes the training dataset. The training process is performed once during the development of the system. A new training database only becomes necessary if a new set of drawings is to be added to the recognition system.

Once the classifier has been trained, the SVM is ready to be used in real-time to classify the drawings sketched by the user. The recognition process can be summarized as follows: (1) Extraction of the contours from each candidate drawing found in pre-processed images; (2) Extraction of the features used by the SVM classifier from each candidate drawing; (3) Flitering of the candidate drawings that have too small areas or a number of vertices outside the range defined for the known drawings; (4) Classification of the candidate drawings using the SVM classifier; (5) Analysis of the SVM output to identify the drawing recognized by the classifier.

For each image frame captured by the camera, the system classifies the hand-drawn sketches found at all segmentation levels resulting from the pre-processing phase. In this way, the system has the classification of the same drawing in different segmentation conditions. Based on these results, the algorithm can search for the best-fitting class. A voting approach is adopted to choose the final classification of the drawing.

3.3. Augmented reality dramatization

The basis of the AR dramatization system is a collection of computer vision techniques used to map the virtual camera according to the real camera, allowing the system to precisely overlay 3D objects onto the video of the real world. The AR dramatization system uses the ARToolKit Library,[2] which encapsulates functions to calculate the position of the real camera based on the size and orientation of physical fiducial markers. Each marker has a distinctive pattern and is associated with a specific location of the virtual world (Fig. 5).

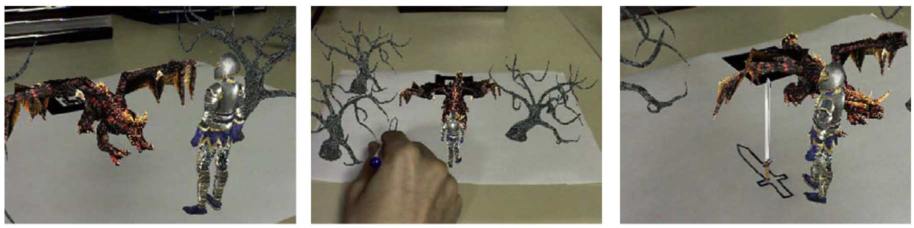
When a known marker is found on the camera image, the virtual objects and characters that are currently on that place in the story are overlapped onto the captured image. The virtual objects are drawn relative to the marker position and orientation. In this way, it is possible to keep them always on the paper surface. In our implementation, we adopt OpenGL as the graphical engine to generate the 3D graphics. The virtual places are modeled by the author of the story, who also specifies the initial position of the virtual characters.

The dramatization system represents the characters of the stories through animated 3D models that can walk freely across the scenarios displayed over the sheets of paper. The story planner conducts the characters' behaviors and the dramatization system represents their actions, controlling and animating the 3D models. The dramatization system allows the virtual characters to perform several parameterized actions, such as "walk", "fight", "talk", "kill", "take", among others. When they need to go to other places (represented by other sheets of papers), they walk outside the current page and are teleported to the other location. The characters then appear in the destination page, walking from the margin of the paper towards their objective. Thus, the paper pages work as separate windows through which the virtual world is revealed. The viewers do not see the entire landscape, but only certain places of interest. However, those visible places form a connected structure, so that to reach a given destination a character may need to traverse through several other places.

When the sketch recognition system identifies a new sketch drawn by the user on a sheet of paper, the dramatization system automatically creates in the virtual world a 3D model of the object represented by the user's sketch. The virtual characters that are in that location are informed about the presence of the new object. The planning system then chooses the appropriate actions for the characters according to the observed situation. All drawings that can possibly be identified by the system are associated with a specific 3D model that is materialized by the dramatization system when a matching sketch is found on the paper. If multiple objects are drawn on the same piece of paper, the system adds all the objects to the virtual world, even when the same object is drawn multiple times. For example, if two swords are near of two knights, each character will pick up one sword. The user interaction process is illustrated in Fig. 6.

The AR dramatization system allows the users to freely move the real camera to visualize the scenes from any angle. However, there are two limitations. First, the fiducial marker must be visible to the camera, so that the system can identify the current location and correctly render the virtual characters and objects over the pa-

Fig. 5. Fiducial markers used by the system to compute the position of the virtual camera according to the real camera. The marker image is also used to identify the virtual places.



(*a*) (*b*) (*c*)

Fig. 6. User Interaction. Image (a) shows a scene being dramatized in a virtual place; image (b) shows the user sketching an object; and finally, image (c) shows the sketched object (a sword, in this case) coming to life in the AR environment.

per. Second, the entire sheet of paper must be visible to the camera when the user draws a new sketch on the paper. After the system recognizes it, the user can again move the camera.

4. Application and evaluation

The prototype application developed to test our system is based on an overly simplified "Swords and Dragons" genre. The virtual world is populated by three main characters: the charming, narcissistic, and lonely princess Marian, who lives under strict protection at a palace; a brave young knight, sir Brian, in love with the princess; and the evil dragon, Draco, constantly waiting for a chance to kidnap the princess. The virtual world is composed of four places: the princess's palace, the dragon forest, a church and the forest where the wandering hero dwells. Users are able to interact with the virtual characters by sketching on a paper. The following six items are currently recognizable and can somehow affect the storyline: a "hand mirror", which may divert the princess's eyes while increasing her level of joy; a "sword", which Brian would gladly wield to supplement his bodily strength; a "shield", adequate for the hero's defense; a "magic stone", that can dangerously increase Draco's strength; a "rat", which can serve either to distract the dragon's attention, or to scare princess Marian; and a "poison bottle", a possible inducement to Marian or Brian to commit suicide in desperate situations. The recognizable sketches used in the prototype are illustrated in Fig. 7.

Considering the opportunities for user interaction, the prototype application is able to generate a considerable number of diversified stories. For example: in more conventional stories, the princess is kidnapped by the dragon and then saved by the hero, who kills the dragon; in stories with a not so happy ending, the hero is defeated by the dragon; and in others with a darker outcome, the dragon kills the princess, or she commits suicide. But the dragon's participation is not really indispensable to cause misfortune. One story was generated wherein the princess, frightened by the rat, broke her mirror, whereupon she became so distraught that she drank the proffered poison.

To evaluate our system, we performed two tests: a technical test to check the performance and accuracy of the system, and then a user evaluation test to check the system's usability from a Human–Computer Interaction (HCI) perspective. The prototype used only 3 characters, 6 objects, and 42 actions, but these numbers do not represent limits to our system. The following sections describe the technical and user evaluation tests.

4.1. Technical evaluation

The technical evaluation concerns the accuracy and the real-time performance of the system. The tests were mainly focused on sketch recognition, which constitutes the most expensive process and includes a machine-learning method that is not guaranteed to provide correct answers at all times.

The evaluation of the sketch recognition system was based on two experiments: (1) the recognition rate test, to check the accuracy of the predicted sketches; and (2) the performance test, to check the time needed to process the input frames and recognize the hand-drawn sketches.

In order to create the dataset to train and validate the sketch recognition system, we manually draw 120 sketches of 6 different classes of objects (20 sketches for each class of object). Based on this set of sketches, we created a collection of 550 pictures by

(*a*) (*b*) (*c*) (*d*) (*e*) (*f*)

Fig. 7. Recognizable sketches. (a) hand mirror; (b) sword; (c) shield; (d) magic stone; (e) rat; (f) poison bottle.

capturing 4 or 5 pictures of each sketch through a conventional webcam from different angles.

For the recognition rate test, we created 5 training datasets ranging from 100 to 300 samples based on the pictures previously captured (each picture generates one training sample). The remaining 250 pictures that were not included in the training datasets were used to validate the sketch recognition system. During the creation of the datasets, we also ensure that pictures from different angles of the same sketch were not included in both training and testing datasets.

To evaluate the performance of the sketch recognition system, we again utilized the collection of 250 pictures, and calculated the average time necessary to perform the pre-processing and the classification of the sketches. The computer used to run the experiments was an Intel Core i7 2.66 GHZ CPU, 8 GB of RAM using a single core to process the algorithms.

Table 1 shows the computed average recognition rate and the result of the performance test, with training datasets ranging from 100 to 300 samples.

Analyzing the test results, it seems fair to conclude that the classifier ensures high recognition rates without sacrificing the system's performance. We can also conclude that the number of samples in the training dataset influences the recognition rate. In our tests, adding more samples increased the recognition rates. However the quality of the samples is more critical than quantity; poor samples can reduce the recognition rates. Also we noticed that the recognition rate is affected by the number of classes (recognizable sketches) in the classifier. More classes result in less accuracy, especially if the drawings have similar shapes. On the other hand, the system performance is not significantly affected by the number of samples or classes. The pre-processing phase is the most time-consuming task; however, the processing time was still relatively low, allowing the process to be executed in real-time. Moreover, the system has a parallel architecture and executes the sketch recognition process on a separated thread, which allows the system to render the output images in real-time without noticeable delay.

4.2. User evaluation

In order to evaluate the system usability, we have conducted a user evaluation with 21 high school students, 16 male and 5 female, aged 16–18 (mean of 17). Eleven of them play video games at least weekly. None of them had previous experiences with interactive storytelling systems.

We asked participants to interact with the proposed system, including objects and changing scenes to influence the story unfolding as they wished. They were asked to interact both with our system (S) and with a modified version of it (M) that used menus to include objects in the scene instead of sketching. In order to reduce learning effects, half of the participants used S first, and the other half used M first.

On average, each session lasted 5 min (standard deviation of 0.87) and included 3.38 interactions (standard deviation of 0.99). In each session, the minimum and maximum number of observed interactions were 1 and 5 respectively. Fig. 8 shows an example of user session summarizing the actions performed by one of the subjects.

After using each version, the participants filled out a questionnaire with 54 questions derived from the IRIS Evaluation Toolkit [31,32]. We evaluate the system usability, the correspondence of system capabilities with user expectations (user satisfaction), the interaction effectiveness and the user experience (curiosity, flow and enjoyment). Each statement was given on a five-point Likert scale ranging from "strongly disagree" (1) through "neutral" (3) to "strongly agree" (5). After having interacted with both versions of the system, the participants were interviewed about their experience.

Table 1
Recognition rate and performance test with training datasets ranging from 100 to 300 samples.

Training samples	Recognition rate (%)	Recognition time (*ms*)
100	92.1	83.6
150	92.8	81.2
200	93.4	84.4
250	93.4	84.7
300	93.8	85.1

Time (*h*:*m*:*s*)	**Event**	**Type**
16:04:03	Story started	System
16:04:05	Princess's palace is visible	System
16:04:31	User started to draw a mirror	Supervisor
16:04:47	User finished drawing the mirror	Supervisor
16:04:49	Object recognition process started	System
16:04:52	**Object recognized: mirror**	System
16:04:53	Marian goes to the mirror	System
16:04:01	Marian gets the mirror	System
16:05:16	Marian dismisses the guards	System
16:05:28	Princess's palace is not visible	System
16:05:32	Hero's forest is visible	System
16:05:45	Hero's forest is not visible	System
16:05:47	Dragon's forest is visible	System
16:05:50	Draco senses that Marian is unprotected	System
16:05:51	Draco goes to the princess's palace	System
16:05:59	Dragon's forest is not visible	System
16:06:05	Princess's palace is visible	System
16:06:16	Draco kidnaps Marian	System
16:06:39	Draco goes to dragon's forest	System
16:06:50	Princess's palace is not visible	System
16:06:53	Dragon's forest is visible	System
16:07:08	Dragon's forest is not visible	System
16:07:11	Hero's forest is visible	System
16:07:23	User started to draw a sword	Supervisor
16:07:40	User finished drawing the sword	Supervisor
16:07:42	Object recognition process started	System
16:07:45	**Object recognized: sword**	System
16:07:45	Brian goes to the sword	System
16:07:53	Brian gets the sword	System
16:07:59	User started to draw a rat	Supervisor
16:08:34	User finished drawing the rat	Supervisor
16:08:40	Object recognition process started	System
16:08:43	**Object recognized: rat**	System
16:08:53	Brian goes to the dragon's forest	System
16:09:01	Hero's forest is not visible	System
16:09:08	Dragon's forest is visible	System
16:09:23	Brian fights against Draco	System
16:09:35	Brian kills Draco	System
16:09:43	Brian saves Marian	System
16:09:55	Story Ended	System

Fig. 8. Example of user session (5:52 min long session, 3 interactions indicated by "Object recognized:", 17 year old student, female). **Time** (***h***:***m***:***s***) indicates the time when the event occurred and **Type** indicates whether the event was recorded by the system or by a human supervisor. Events are automatic translations of the predicates used by the system, such as *go*( ), *get*( ), *visible*( ), ...

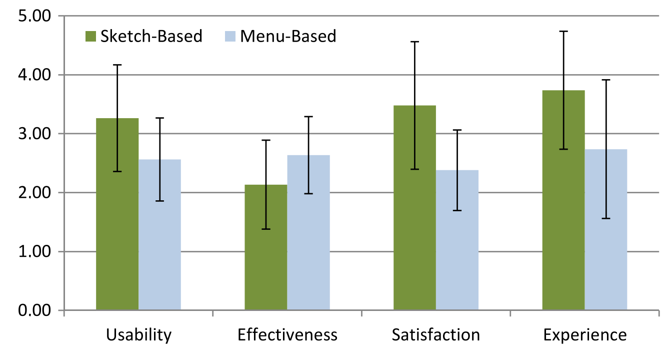


Fig. 9. Average number of points (within a 5-point Likert scale) of the system *usability*, interaction *effectiveness*, user *satisfaction* (*i.e.* user expectations), and user *experience* (curiosity, flow, and enjoyment), with error bars indicating standard deviation around the mean, for the two versions of the system (*sketch-based* and *menu-based*).

Fig. 9 summarizes the results from the questionnaires. The menu-based version of the system produces slightly better effectiveness when compared with the sketch-based version, probably because of some limitations of the sketch recognition algorithm. On the other hand, the sketch-based version of the system clearly increases the user satisfaction, improved the user experience and the usability of the system. As far as the interviews are concerned, all participants stated that they preferred to interact with the sketch-based version, because it was more interesting, attractive, exciting, and allowed them to enhance their senses of participation and immersion, despite the slightly increased effort, mostly due to some limitations of the recognition algorithm.

The recognition rate during the user evaluation test was 83.3%, which is less than the one obtained during the technical evaluation. Observing the cases where the drawings were not correctly recognized, we concluded that most part of the mistakes occurred because some users tried to draw sketches substantially different from the ones that the system is capable of recognizing. We also observed that in 63.3% of the cases of incorrectly recognition, the users tried to draw the objects again and they were correctly recognized in the second attempt. In the others cases, the users did not tried to draw the objects again, they simply let the story continue.

5. Conclusion

In this paper, we presented a mixed reality interactive storytelling system that allows users to visualize stories in AR and to interact with virtual characters by sketching objects on a sheet of paper. We argue that the combination of a mixed reality visualization interface with a sketch-based interaction interface can offer an attractive environment for developing interactive narratives. The results from our user study showed that the use of hand drawings as a form of interaction improves user satisfaction and experience and the system usability. As far as we are aware, this is the first time a pencil-and-paper interactive storytelling system is implemented and tested.

We are currently working on an extended version of the system with more objects. Also we are planning a more in-depth user evaluation. As an ongoing work, we are starting exploring scenario changes (like drawing dark clouds that may cause rain) and other media like video-based dramatization.

Acknowledgments

This work was partially supported by CAPES (Coordination for the Improvement of Higher Education Personnel, linked to the Ministry of Education) under grant RH-TVD 01/2007 No. 133/2008, and by FINEP (Brazilian Innovation Agency) and CNPq (National Council for Scientific and Technological Development) belonging to the Ministry of Science and Technology. Also we would like to thank the anonymous reviewers for the valuable suggestions and meticulous corrections.

References

[1] N. Hagbi, R. Grasset, O. Bergig, M. Billinghurst, J. El-Sana, In-place sketching for content authoring in augmented reality games, in: Proceedings of IEEE Virtual Reality Conference, 2010, pp. 91–94.
[2] A. Vogelsang, B. Signer, The lost cosmonaut: an interactive narrative environment on the basis of digitally enhanced paper, in: Proceedings of 3rd International Conference on Virtual Storytelling, 2005, pp. 270–279.
[3] D. Kuka, O. Elias, R. Martins, C. Lindinger, A. Pramböck, A. Jalsovec, P. Maresch, H. Hörtner, P. Brandl, Deep space: high resolution VR platform for multi-user interactive narratives, in: Proceedings of the 2nd Joint International Conference on Interactive Digital Storytelling, 2009, pp. 185–196.
[4] V. Bayon, J.R. Wilson, D. Stanton, A. Boltman, Mixed reality storytelling environments, Virtual Reality 7 (1) (2003) 54–63.
[5] S. Dow, M. Mehta, A. Lausier, B. MacIntyre, M. Mateas, Initial lessons from AR-façade, an interactive augmented reality drama, in: ACM SIGCHI Conference on Advances in Computer Entertainment, 2006.
[6] M. Mateas, Interactive drama, art, and artificial intelligence (Ph.D. thesis), School of Computer Science, Carnegie Mellon University, Pittsburgh, 2002.
[7] M. Cavazza, F. Charles, S.J. Mead, O. Martin, X. Marichal, A. Nandi, Multimodal acting in mixed reality interactive storytelling, IEEE Multimedia 11 (3) (2004) 30–39.
[8] D.T. Huynh, K. Raveendran, Y. Xu, K. Spreen, B. MacIntyre, Art of defense: a collaborative handheld augmented reality board game, in: Proceedings of the ACM SIGGRAPH Symposium on Video Games, 2009, pp. 135–142.
[9] J. Leitner, C. Köffel, M. Haller, Bridging the gap between real and virtual objects for tabletop game, International Journal of Virtual Reality 7 (3) (2009) 1–5.
[10] P. Ekman, W.V. Friesen, Constants across cultures in the face and emotion, Journal of Personality and Social Psychology 17 (1971) 124–129.
[11] V. Vapnik, The Nature of Statistical Learning Theory, Springer, New York, 1995.
[12] I. El-Naqa, Y. Yang, M.N. Wernick, N.P. Galatsanos, R.M. Nishikawa, A support vector machine approach for detection of microcalcifications, IEEE Transactions on Medical Imaging 21 (12) (2002) 1552–1563.
[13] T. Cai, T. Hofmann, Hierarchical document categorization with support vector machines, in: Proceedings of the 13th Conference on Information and, Knowledge Management, 2004, pp. 78–87.
[14] E.S. Lima, C. Pozzer, M. Ornellas, A. Ciarlini, B. Feijó, A. Furtado, Virtual cinematography director for interactive storytelling, in: Proceedings of the International Conference on Advances in Computer Entertainment Technology, 2009, pp. 263–270.
[15] M.K. Hu, Visual problem recognition by moment invariants, IRE Transactions on Information Theory 8 (1962) 179–187.
[16] Z. Zhou, A.D. Cheok, J. Tedjokusumo, G.S. Omer, wIzQubes – a novel tangible interface for interactive storytelling in mixed reality, International Journal of Virtual Reality 7 (4) (2008) 9–15.
[17] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, B. MacIntyre, Recent advances in augmented reality, IEEE Computer Graphics and Applications 21 (6) (2001) 34–47.
[18] D.W.F. van Krevelen, R. Poelman, A survey of augmented reality technologies, applications and limitations, International Journal of Virtual Reality 9 (2) (2010) 1–20.
[19] A. Jaimes, N. Sebe, Multimodal human-computer interaction: a survey, Computer Vision and Image Understanding 108 (1–2) (2007) 116–134.
[20] A. Malkawi, R. Srinivasan, Multimodal human–computer interaction for immersive visualization: integrating speech–gesture recognitions and augmented reality for indoor environments, in: Proceedings of the 7th IASTED Conference on Computer Graphics and, Imaging, 2004, pp. 171–175.
[21] S. Dow, M. Mehta, E. Harmon, B. MacIntyre, M. Mateas, Presence and engagement in an interactive drama, in: Proceedings of SIGCHI Conference on Human Factors in Computing Systems, 2007, pp. 1475–1484.
[22] M. Cavazza, J.-L. Lugrin, D. Pizzi, F. Charles, Madame bovary on the holodeck: immersive interactive storytelling, in: Proceedings of the 15th International Conference on Multimedia, 2007, pp. 651–660.
[23] T. Igarashi, B. Zeleznik, Guest Editors' introduction: sketch-based interaction, IEEE Computer Graphics and Applications 47 (1) (2007) 26–27.
[24] I.E. Sutherland, Sketchpad: a man-machine graphical communication system, in: Proceedings of AFIPS, 1963, pp. 329–346.
[25] O. Bergig, N. Hagbi, J. El-Sana, M. Billinghurst, In-place 3D sketching for authoring and augmenting mechanical systems, in: Proceedings of the 8th IEEE International Symposium on Mixed and Augmented Reality, 2009, pp. 87–94.
[26] Y Cai, C. Miao, A.-H. Tan, Z. Shen, A hybrid of plot-based and character-based interactive storytelling, in: Proceedings of the 2nd International Conference on Technologies for E-learning and Digital Entertainment, 2007, pp. 260–273.
[27] A.E.M. Ciarlini, C.T. Pozzer, A.L. Furtado, B. Feijo, A logic-based tool for interactive generation and dramatization of stories, in: Proceedings of the International Conference on Advances in Computer Entertainment Technology, 2005, pp. 133–140.

[28] A.E.M. Ciarlini, M.M. Camanho, T.R. Doria, A.L. Furtado, C.T. Pozzer, B. Feijo, Planning and interaction levels for TV storytelling, in: 1st Joint International Conference on Interactive Digital Storytelling, 2008, pp. 198–209.

[29] F.G.A. Silva, A.E.M. Ciarlini, S.W.M. Siqueira, Nondeterministic planning for generating interactive plots, in: Proceedings of the 12th Ibero-American Conference on AI, 2010, pp. 133–143.

[30] M. Cavazza, F. Charles, S. Mead, Character-based interactive storytelling, IEEE Intelligent Systems 17 (4) (2002) 17–24.

[31] C. Klimmt, C. Roth, I. Vermeulen, P. Vorderer, The empirical assessment of the user experience in interactive storytelling: construct validation of candidate evaluation measures, Technical Report (2010). Integrating Research in Interactive Storytelling. Report number D7.2, pp. 1–35, Available at: http://www.scm.tees.ac.uk/f.charles/fp7-iris/deliverables/IRIS-FP7-ICT-231824-Deliverable-7.2.pdf.

[32] C. Roth, P. Vorderer, C. Klimmt, The motivational appeal of interactive storytelling: towards a dimensional model of the user experience, in: International Conference on Interactive Digital Storytelling, 2009, pp. 38–43.

[33] A. Jhala, C Rawls, S. Munilla, R.M. Young, Longboard: A Sketch Based Intelligent Storyboarding Tool for Creating Machinima, in: Proceedings of the Florida Artificial Intelligence Research Society Conference (FLAIRS), 2008, pp. 386–390.

[34] J. Jorge, F. Samavati (Eds.), Sketch-Based Interfaces and Modeling, Springer, 2011.