CamBridge: A Bridge of Camera Aesthetic between Virtual Environment Designers and Players

Chanchan Xu
Communication University of China
Chaoyang Qu, Beijing Shi, China
cc129xu@126.com

Guangzheng Fei
Communication University of China
Chaoyang Qu, Beijing Shi, China
gzfei@cuc.edu.cn

Honglei Han
Communication University of China
Chaoyang Qu, Beijing Shi, China
hanhonglei@sina.com

ABSTRACT

The designer of the virtual environment have been trying for decades to provide the player with more enjoyable, comfortable and also informative user experiences, and yet still fail to ensure that the player follow the preset instructions and even implicit suggestions faithfully and naturally, due to the designer's invisibility during runtime, and the player's individual diversity and individual impromptu in manipulations. We believe that the camera is the mainly messenger for the designer and the player to communicate, and intend to build a bridge between them. By binding the designer's aesthetic ideas to the parameters of the camera's movement, we enable the player to roam in the virtual scene with the guidance from the designer. We also propose a navigation guiding language (NGL) to assist the binding and the guiding process. A user study is made to evaluate the performance of our method. Experiments and questionnaires have shown that our method can offer a more attentive and pleasing experience to the player with implicit guidance.

CCS CONCEPTS

• **Human-centered computing → Walkthrough evaluations**; *Virtual reality*; • **Applied computing → Computer-aided design**;

KEYWORDS

camera control, implicit guidance, camera aesthetic, interactive arts

ACM Reference Format:

Chanchan Xu, Guangzheng Fei, and Honglei Han. 2018. CamBridge: A Bridge of Camera Aesthetic between Virtual Environment Designers and Players. In *International Conference on Virtual Reality Continuum and its Applications in Industry (VRCAI '18), December 2–3, 2018, Hachioji, Japan.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3284398.3284423

1 INTRODUCTION

The paradoxical relationship between designers and players has been investigated by many researchers for years. People often realize the large difference between the designer's original intention and the player's actual experiences. Communication gap between the designer and the player can hardly be filled due to the non-simultaneousness of their activities. Especially, in a virtual environment application, the designer can only design an environment and provide a possible experience according to his assumption of the player's behavior. In this paper, we intend to build a bridge of camera aesthetic-a CamBridge to connect the two and provide the player with an autonomous manipulation under the designer's guidance.

The virtual camera is widely considered to play a vital role in conveying information to the player and influencing the user experience. It offers the window for the player to observe the virtual environment and transfers the feedback of the player, i.e. the camera manipulations, to the program for reaction. Acting as a special filmic symbol, the position, orientation and movement of the virtual camera make the foundation of narrative context. Photographers, designers and directors of traditional non-interactive arts, such as photography, cinematography and animation, often express their intention in a picture or a story through the view of a carefully placed camera or the rendering of a series of delicately arranged cameras.

New challenges have been brought out since the arising of interactive arts especially the ones including interacting with 3D virtual environments, such as interactive storytelling and video games. The interactivity gives the player the permission to control his views and positions through the virtual camera by himself, which often leads to bad or un-recommended views which may easily cause confusions or even lost especially when the environment is complicated and the player is a novice. Besides, people found it inappropriate for the artist to carry the creative power to their work only in the design stage. The artists are eager to show the audience their own part of this creative process during the player's observation process, which is unlikely to realize because of the player's individual difference, individual diversity and individual impromptu in manipulating the virtual camera. To predict all the possible interactions of the player so to predefine a proper reaction is a huge and nearly impossible task for the designer to accomplish.

The designer possesses more aesthetic accomplishment and also more cognitions of the observed scene than common players. His willing is expected to be a great help in improving the operation experience of the player, and decreasing the confusion and lost situations. In this paper, we try to find an approach to implicitly insert the designer's design philosophy into the user's operations, and build a CamBridge for the designer and the player to communicate. Our propose is to enable the player not only obtain visual pleasant views which the designer eagerly wants to show, but also enjoy a relatively free manipulation while interacting with the virtual environment.

VRCAI '18, December 2–3, 2018, Hachioji, Japan

ACM ISBN 978-1-4503-6087-6/18/12…$15.00
https://doi.org/10.1145/3284398.3284423

The contributions of our work are as follows:

(1) Propose a camera control method by binding the designer's intension, ideas or thoughts to the virtual camera to further influence the player's roaming behaviors with guidance.

(2) Put forward a novel descriptive language, namely NGL, to assist the binding and the guiding process.

(3) Employ a user study to evaluate the performance of our method.

In the following parts, we first introduce the former research works, and then describe our camera control method with the designer's guidance. Our NGL is illustrated in detail in section 4. Section 5 shows the experiments with survey. In the final section, we conclude the paper, discuss the limitations and state some future works.

2 RELATED WORK

Camera control in 3D virtual environments holds a great promise in the area of 3D games, education, science etc., and has been a hot topic for years. Below we introduce the state of the art related to this topic.

2.1 Automatic Camera Control

The concept of camera control was proposed by Gleicher[Gleicher and Witkin 1992], regarding it as a process to manually control the virtual camera by confining a number of properties. The automatic camera control tries to define an abstraction layer that permits to control the camera through high-level and environment-independent rules, which are later translated into camera movements by a camera controller[Burelli 2012]. Olivier formalized the problem of camera control as an optimization problem which involves a set of image properties[Olivier et al. 1999]. Galvane[Galvane et al. 2015] automatically generated camera paths through refining a rough path created by a analyse of characters motion and user-defined framing properties. Joubert [Joubert et al. 2016] placed the camera though some visual composition principles such as the rule of thirds, for showing safer quadrotors.

2.2 Cinematography and interactive narrative

Camera movements are used as special codes to tell the stories through a series of camera motions and frame shifts, which make them useful in the narrative of games and other applications.

Li [Li and Xiao 2005] employed three modules which are director, cinematographer and editor, in their system to determine the camera configurations and extract the stylistic parameters of the cinematography idioms to determine the aesthetic style. Amerson presented a system composed of translator, director and cinematographer [Amerson et al. 2005]. They created a scene tree as the camera control architecture and use a FILM language to code the scene and shot in the scene tree. Before that, Christianson devised a DCCL language to describe the shot series for narrative [Christianson et al. 1996]. Jhala [Jhala and Young 2005] tried to formalize the film language for cinematic narratives presentations.

Some researchers intended to incorporate the authors' intend to the narrative process for allowing the designer's guidance on the story, such as paper [Hughes and Lewis 2000; Peinado and Gervás 2004]. Other researchers allowed the player to discover his own plot[Ramirez and Bulitko 2015; Yu and Riedl 2012]. These methods have obvious advantages in expressing the content of an environment to the user, but ignore the interaction between the user and the designer, which still remains a huge challenge.

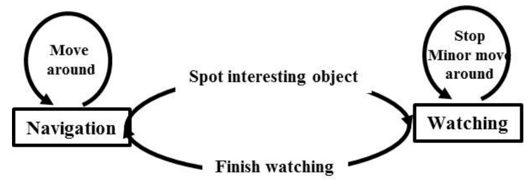


Figure 1: User behavior in 3D virtual environment

The basic goal in the applications of games and virtual scene walkthrough is to provide the player with enough information in order to help them get a clear conception of the whole scene. However, the player may often concentrate on irrelevant objects and miss the important clues in actual operations. We intend to develop an interactive tool to help the designer guide the behavior of the player. The guidance happens implicitly without much impacting the player's freedom.

3 CAMERA CONTROL WITH GUIDANCE

3.1 The Guiding Theory

When roaming in a virtual 3D environment, we observe that the user's interaction behaviors mainly follow two processes: navigating in the environment and viewing a scene spot. The player may move around in the virtual scene until he spots an interesting object, then he stops and moves in minor distance to adjust his view to watch this object. After some period, the player moves on until he meets the next interesting object.

The so-called interesting object can either be a scenic building, a beautiful vase or a painting. During the watching process, the observed object and the viewing angle affects the user's experience greatly. Therefore, we give the designer the right to set the position of the interesting spot and also specify the direction of the view from which the designer believe the user can obtain a better picture and a clearer conception about the interesting object. The interesting spot here is defined as an "Aiming-Point" and the view specified by the designer is called "Best-View". Below we will illustrate the structure of them.

3.2 The Guiding Structure

Being recommended by the designer, the aiming-point is believed to own more saliency than other objects in the environment and the best-view is also considered to be prior to the common user's viewpoints. The saliency is defined as the region which has significant differences with the surrounding area [Koch and Ullman 1985]. Therefore, the aiming-point here is considered as an attraction point which attracts the attention of the player. The attraction strength is effective in a certain range, and falls off as the distance to the aiming-point increases. The best-view is represented as a camera's orientation and position. One aiming-point can possess more than two best-views.

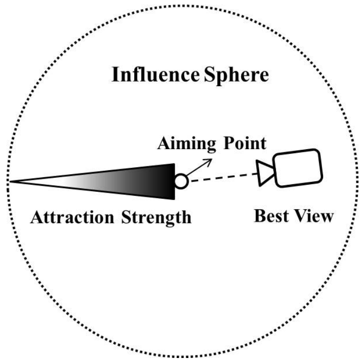


Figure 2: Aiming-point and best-view

3.3 The Guiding Procedure

As we mentioned before, the aiming-points and best-views are preset by the designer, and are then transformed to the virtual camera's parameters. The whole steps of camera guidance are: 1) Pick one aiming-point; 2) Generate guiding curves from the user's position to the selected aiming-point; 3) Guide the user's path and direction according to the generated curve; 4) Detect the collision.

3.3.1 Picking One Aiming-point. To pick one aiming-point from the aiming-point set depends on its influence range and the attraction strength. We assume that the player can only focus on one aiming-point at a time. When coming across more than two aiming-points, the picking strategy is proceeded according to the player's viewing history, his own preferences and the concerned aiming-points' attraction range and strength. The formulation is as follows:

$$s = f * \alpha * (1 - \gamma) \tag{1}$$

where, f represents the aiming-point's attraction strength, α is a preference weight, γ is the attenuation rate. According to the distance decay regularity, the strength is inversely proportional to the square of distance. Therefore, we define $\gamma = d^2/r^2$ and rewrite equation 1 as:

$$s = \frac{f * \alpha * (1 - d^2)}{r^2} \tag{2}$$

where, d is for the Euclidean Distance between the aiming-point and the user camera, r is the influence range radius of the aiming-point. The player tends to pick the one which is unvisited and whose attraction strength is larger to watch. Figure 3 shows a situation to pick between two aiming-points.

3.3.2 Generation of Guiding Curve. Once the aiming-point has been selected, the activated aiming-point is then used to influence the user's roaming behavior. A cubic Hermite curve is created as the guiding path which is applied to guide the user's travel trend.

The formulas are as follows:

$$H = \begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \tag{3}$$

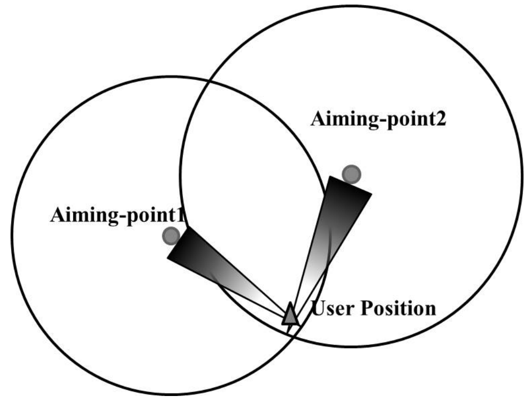


Figure 3: Picking between two aiming-points

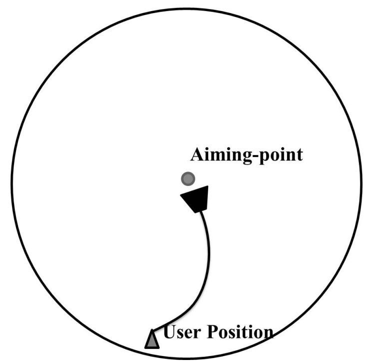


Figure 4: Generation of guiding curve

$$S = \begin{pmatrix} s^3 & s^2 & s^1 & s^0 \end{pmatrix}^T \tag{4}$$

$$C = \begin{pmatrix} P_1 & P_2 & T_1 & T_2 \end{pmatrix}^T \tag{5}$$

where, H is the Hermite matrix, S is the difference vector and C is the parameter vector. The range of the parameter s is from 0 to 1. P_1, P_2, T_1, T_2 in vector C represent the start point position, the end point position, the starting tangent and the ending tangent. The current user position P in the Hermite is calculated as follows:

$$P = S * H * C \tag{6}$$

In order to get a proper guiding path, we adjust the vector of the start point and the end point in the Hermite curve.

The start point and the starting tangent individually represent the position and the direction when the user enters the influence area of the aiming-point, while the endpoint and the ending tangent individually represent the position and the direction of the best camera for the aiming-point. The starting tangent and the ending tangent take great effect in shaping the Hermite curves. It is believed that the user can stand a large range of view diversion at the beginning, but cannot tolerate an apparent view change when get closer to the destination. In this way, an adjustment is made in order to get a smooth transition to relieve the uncomfortable feelings. The formulations are defined as follows:

$$|T_1| = |P_1 - P_2| * w_1, \quad (1 < w_1 < 1.5) \tag{7}$$

$$|T_2| = |P_1 - P_2| * w_2, \quad (0 < w_2 < 1) \tag{8}$$

where, same as before, P_1 and P_2 represent the position of the start point and the end point, T_1 and T_2 stand for the starting tangent and the ending tangent. w_1 and w_2 are the weight to change the length of T_1 and T_2. Experiments show that this adjustment can get a smoother curve with less double points, sharp points and break points.

3.3.3 Attentive Guiding. Inspired by the implementation of steering behaviors for autonomous characters in paper [Ondřej et al. 2010; Reynolds 1999], we use an agent-based algorithm. The basic processes are as follows:

(1) Calculate the deflection force f to be imposed on the camera steer according to the position of the user for each frame.

$$f' = (P_t - P_n) * t \tag{9}$$

where P_t is the target position and P_n is the current position of the player. The target position is calculated by projecting the current position of the user to the guiding curve, while the current position is manipulated by the user. t is the delta time between two continuous frames.

(2) Update the orientation and position of the camera steer with the deflection force to fit the direction of the guiding path.

$$V' = V_f * \beta + V * (1 - \beta), \quad (0 < \beta < 0.2) \tag{10}$$

where, V' represents the direction after the modification; V_f is the force vector which is imposed on the camera steer; V records the direction of the user at present and β is an empirical value ranges from 0 to 0.2.

3.3.4 Collision Detection. Once the user runs into the influence area, i.e. the bounding box of an obstacle, the guiding process employs a deflection force to the user camera. The deflection force is perpendicular to the camera speed and is calculated and activated at every frame until the user bypasses the obstacles. When the avoidance process is finished, the Hermite curve is recalculated.

4 DESIGN OF NGL

To record the editing operation of the designer, we design a Navigation Guiding Language (NGL) for our program. The NGL should be expressive enough to describe the aesthetic edits of the designer. These edits are then translated into constraints or rules to confine the user's behaviors to some tolerated extent. A reasonable NGL should not only provide the designer a visualized interface for editing, but also can influence the user's behaviors without being noticed. It is a visualization tool for the designer, an implicit influence for the user and also an understandable programming language. This is our purpose for NGL design.

4.1 Grammar of NGL

Our NGL contains vocabulary, syntax and semantics so as to form a complete system (see Figure 5).

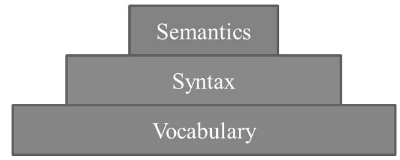


Figure 5: The structure of NGL

The vocabulary includes all the proper words which may be used in the description of NGL. According to the usage of the words, the basic vocabulary can be divided into three types: objects, attributes and data type. Considering that our system is applicable for 3D virtual scene roaming, the NGL should take in a complete description of the objects in the virtual scene.

We believe that users usually have some interesting points while observing a scenario and pay more attention to these interesting points. These points can greatly influence the behaviors and routes of the user, named aiming-points. While observing an aiming-point, the user used to choose a comfortable viewing angle, called as best-view. Thus, we defined the "objects" as follows: scene, object, aiming-point and best-view. The attributes of scene are defined as: *ID*, *Desc*, *Position*, *Size*, *GroupNo*, *AimNo*, *CamNo*, *Group*, *ObjectList*, *AimingPoint* and *BestView*. Among these attributes, the attribute of Group contains the ids of those object members which are considered as a whole in the scene; the attribute of *ObjectList* includes the basic information of the objects such as *Position*, *BoundBox*, *IsColliOn*; the attribute of an AimingPoint describes the position, weight, range, influence force value, isactivated and the corresponding camera to represent the best-view; the attribute of BestView represents the position, the focus position, and the field of view of the best camera. The data types can be mainly divided into "value type" and "strings". Value type is demarcated with "#" and can be divided into integer(N), float(F) and bool(B), while a string is demarcated with "$", consisting of a number of characters.

In syntax, we define a sentence which starts with "<" and end with "/>" as a valid complete sentence in our system and a number of sentences which begin with "<*TypName*>" and end with "</*TypName*>" as a valid compound sentence. The "*TypName*" can be replaced by *BaseInfo*, *Group*, *ObjectList*, *Object*, *AimingPoint*, *BestView*, *Scene* etc. Indentation can help increase the readability of NGL.

Semantic definitions are as follows: "*Scene*" represents the whole virtual scenario; "*Group*" means a number of objects which work as a whole; "*ObjectList*" is a description list for scene object; "*Object*" is a visual independent component of the virtual scene. Figure 6 gives an example of NGL.

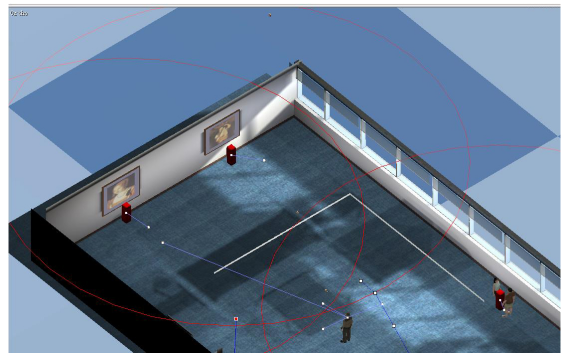
4.2 Visualization of NGL

We develop an interactive interface for the designer with which all the manipulations of the designer can be recorded as a file written in NGL. The interface may include 3 modes: free mode, aiming-point mode and best-view mode, corresponding to the editing of scene, aiming-point and best camera (see in Figure 7).

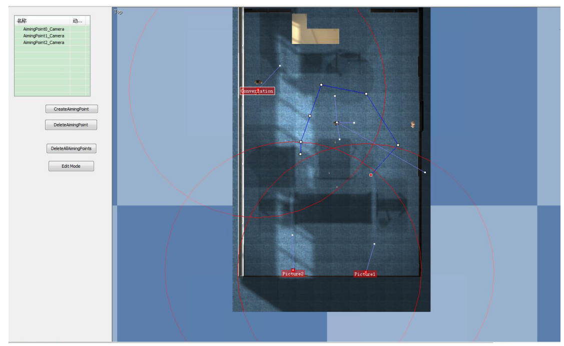
Under the free editing mode, a perspective view is offered for the designer to freely change the view and observe the whole scene. Under the aiming-point mode, the view is fixed to orthogonal view

```
<Scene      ID = $String>
    <BaseInfo>
    <Desc   description=$String>
        <Position       x=#F  y=#F  z=#F />
        <Size           x=#F  y=#F  z=#F />
        <GroupNo        groupno=#N />
        <AimNo          aimno=#N />
        <CamNo          camno=#N />
    </BaseInfo>
    <Group1             ID = $String>
        <MemNo          menno=#N/>
        <Member1        ID = $String/>
        ...
    </Group1>
    <ObjectList>
        <MemNo          menno=#N/>
        <Object1        ID = $String/>
            <Desc       description = $String/>
            <Position   x=#F  y=#F  z=#F />
            <BoundBox x=#F  y=#F  z=#F />
            <IsColliOn  iscollideon = #B />
        </Object1>
        ...
    </ObjectsList>
    <AimingPoint1       ID=$String>
        <Position       x=#F  y=#F  z=#F />
        <Weight         weight = #F/>
        <Range          range = #F/>
        <ForceV         influforcevalue = #F />
        <IsActived      isact = #B/>
        <Camera         ID = $String/>
    </AimingPoint1>
    ...
    <BestView1          ID = $String>
        <Postion        x=#F  y=#F  z=#F />
        <Focus          x=#F  y=#F  z=#F />
        <FOV                        fov = #N/>
        <IsAnimated     isani = #B/>
    </BestView1>
    ...
</Scene>
```

Figure 6: An example of NGL



(a) Free mode



(b) Aiming-point mode

(c) Best-view mode

Figure 7: Three editing mode in NGL visualization

and the designer can use a control panel to set, edit and delete an aiming-point. We provide visualized tools for this editing. For example, the designer can set the aiming-point by creating a cube at the intending place, and change the influence area by just clicking the aiming-point and roll the mouse wheel. The best-view of a camera is set through a camera.

Once the designer determined the aiming-points and their best-views, both the positions and orientations of them will be recorded in NGL and stored in a file which will be further loaded during the player's operation time.

We try to design the NGL as an abstract, high-level and visualized language, with friendly tool interface for designers, both proficient and non-proficient, to help them design the scenes with high quality and speed. Unlike the traditional ways, the virtual camera is no longer a simple tool which only records frames of lens, positions and shot scales, but as a communication bridge between the designer and the user. It will convey an implicit influence during the interaction of the user and the application, guiding the user's behaviors to fit the designer's expectation.

Table 1: Information of aiming-points

ID	Desc	Best-view	w	range
A	Conversation	Figure 8(a)	0.5	100
B	Picture 1	Figure 8(b)	0.5	100
C	Picture 2	Figure 8(c)	0.5	100

5 EXPERIMENTS AND SURVEY

5.1 Experiments and Results

We implement our method in a virtual environment, using Visual Studio 2005 as the development platform and the Virtual Reality Platform as the running platform. We predefine three aiming-points which are shown in Figure 8. Details and parameters are illustrated in Table 1.The corresponding NGL is shown in Figure 9.

The aiming-point is specified with a box and the best-view is indicated by a camera icon. The guiding process is shown in Figure 10.

The influence ranges in the parameters of Table 1 are all set to 100 and their strengths are equal to each other. If we change the influence range of aiming-point C to be 200, then the system will first guide the player to point C, i.e. where the picture 1 is(see Figure 11).

5.2 User Study

For further evaluate the effectiveness and applicability of our method, we performed a user study. The virtual environment is set as above, a virtual museum with three aiming-points, each attached with one best-view. We designed a questionnaire for survey. The content of the questionnaire is shown as follows.

(1) Have you ever noticed the guidance of the system?

A: Yes; B: No.

(2) How many places did the system try to guide you (among the three views in Figure 8)?

A: Three; B: Two; C: One; D: None

(3) Did the guidance have impact on your freedom of manipulation?

A: None; B: Rarely; C: Some; D: Serious

(4) Can you bear the influence which the guidance has on you?

A: Totally; B: Mostly; C: Rarely; D: Not at all.

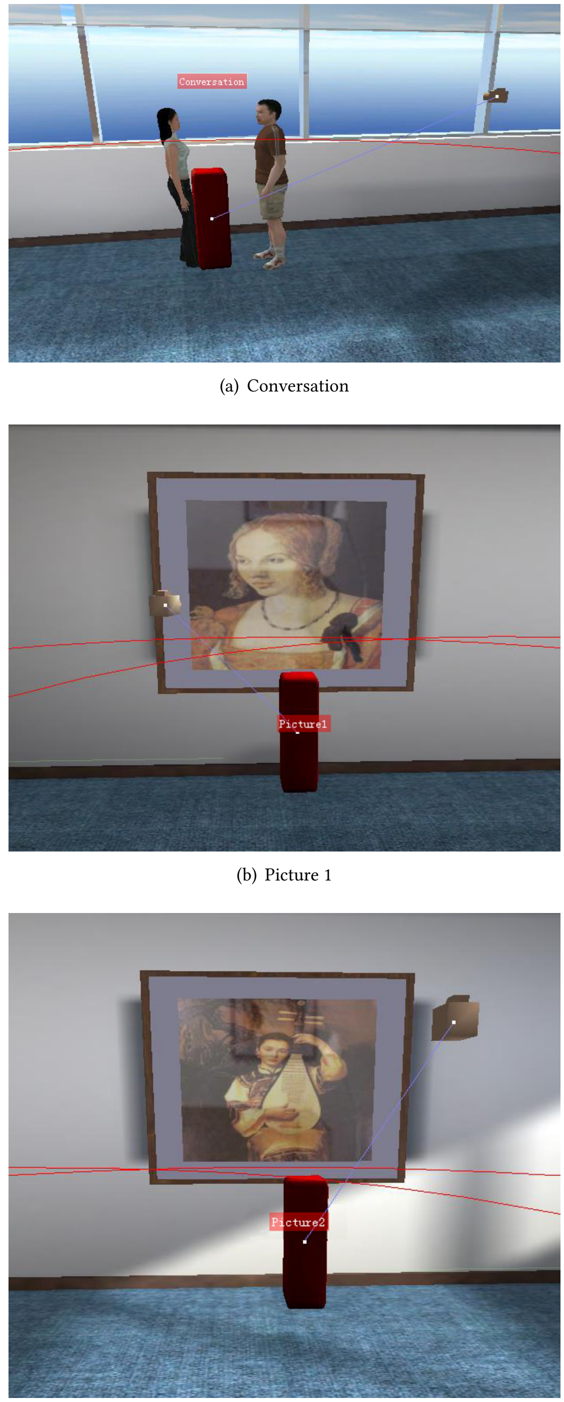
(5) Was it comfortable for you to roam under the guidance?

A: Totally; B: Mostly; C: Rarely; D: Not at all.

(6) Has it helped in your comprehension of the environment?

A: Totally; B: Mostly; C: Rarely; D: Not at all.

The first question is used for judging the effectiveness of our methods in a general aspect. People who haven't notice the guidance can just skip the following questions. The second to the sixth questions value the degree of satisfaction about how the player feels about our program during his roaming in the virtual environment. Especially, the second question records the noticed aiming-points number; the third question measures the interruption degree of the guidance; the forth question qualifies the tolerance of our guidance's interruption; the fifth question asks the comfortable degree of the player during his roaming; and the last question surveys the use in helping strengthen the player's understanding ability about



(a) Conversation

(b) Picture 1

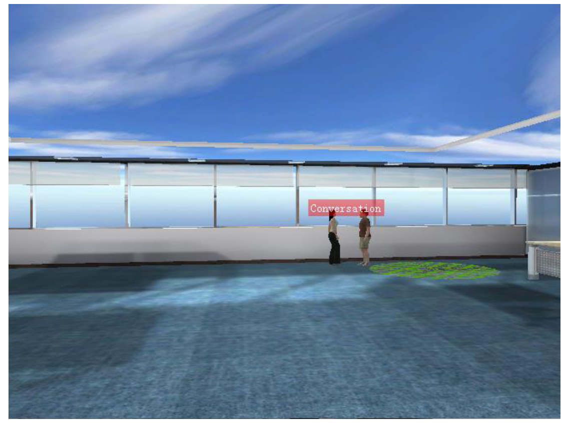
(c) Picture 2

Figure 8: Best-views and aiming-points

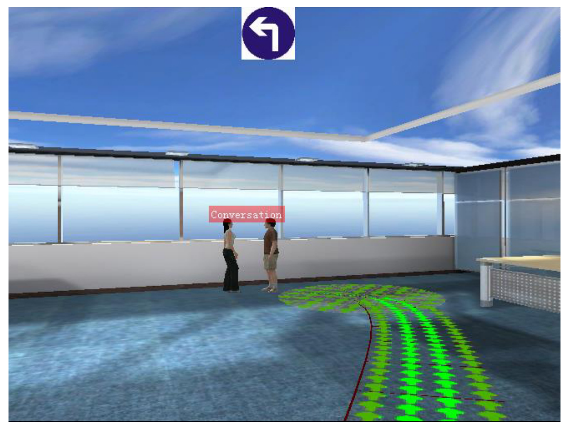
```
<Scene      ID = "$Scene1">
    <BaseInfo>
    <Desc  description="$Virtual Museum">
        <GroupNo       groupno="#N0" />
        <AimNo         aimno="#N3" />
        <CamNo         camno="#N3" />
    </BaseInfo>
    <AimingPoint1      ID="$AimingPoint1">
        <Position      x="#F-56.97"  y="#F0.03"  z="#F-12.72" />
        <Weight        weight = "#F1.0"/>
        <Range         range = "#F100.0"/>
        <ForceV        influforcevalue="#F0.73"/>
        <IsActived     isact = "#B1"/>
        <Camera        ID = "$BestView1"/>
    </AimingPoint1>
    <AimingPoint2      ID="$AimingPoint2">
        <Position      x="#F27.42"  y="#F0.12"  z="#F-155.04" />
        <Weight        weight = "#F1.0"/>
        <Range         range = "#F100.0"/>
        <ForceV        influforcevalue="#F0.49"/>
        <IsActived     isact = "#B1"/>
        <Camera        ID = "$BestView2"/>
    </AimingPoint2>
    <AimingPoint3      ID="$AimingPoint3">
        <Position      x="#F29.06"  y="#F0.12  z="#F-153.49" />
        <Weight        weight = "#F1.0"/>
        <Range         range = "#F100.0"/>
        <ForceV        influforcevalue="#F0.18"/>
        <IsActived     isact = "#B1"/>
        <Camera        ID = "$BestView3"/>
    </AimingPoint3>
    <BestView1         ID = "$BestView1">
        <Postion       x="#F-40.36"  y="#F15.65"  z="#F5.26" />
        <Focus         x="#F-56.97"  y="#F0.03"  z="#F-12.72" />
        <FOV                  fov = "#N60"/>
        <IsAnimated    isani = "#B0"/>
    </BestView1>
    <BestView2         ID = "$BestView2">
        <Postion       x="#F34.02"  y="#F15.65"  z="#F-132.95" />
        <Focus         x="#F27.42"  y="#F0.12"  z="#F-155.04" />
        <FOV                  fov = "N60"/>
        <IsAnimated    isani = "#B0"/>
    </BestView2>
    <BestView3         ID = "$BestView3">
        <Postion       x="#F-29.63"  y="#F15.65"  z="#F-126.02" />
        <Focus         x="#F29.06"  y="#F0.12  z="#F-153.49" />
        <FOV                  fov = "#N60"/>
        <IsAnimated    isani = "#B0"/>
    </BestView3>
</Scene>
```

Figure 9: The NGL for virtual museum

the environment. We invite 8 people to attend the survey. These people are graduated students, and PhDs whose majors are either animation design or game design, which means that they possess the basic knowledge about how to roam in the virtual environment. Before running the program, they were not told anything about the real purpose of our survey. The questionnaire was given to them when they finished their roaming. We mark the points of the four answers of question 2 to question 6 from 4 to 1, and no points for question 1, as the first question mainly acts as a filter to keep those



(a) Roam around



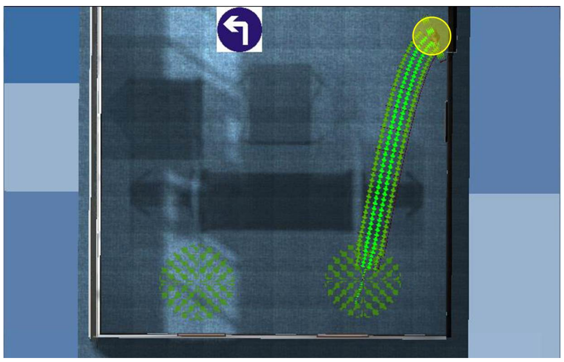
(b) Generate guiding path

(c) Guide to the best-view

Figure 10: Guiding process

further used questionnaires to analyze. However, when the player didn't feel the guidance, i.e. replied no in question 1, that means our method is failed for him. We collect all the questionnaires and calculate the scores which are shown in Table 2.

(a) Influence range 100,100,100 for Aiming-point A,B,C



(b) Influence range 100,200,100 for Aiming-point A,B,C

Figure 11: Comparison of different influence range

Table 2: Scores of the questionnaire

Q ID	MaxScore	TotalScore	AvgScore
2	4	31	3.87
3-6	16	96	12

All the eight people have declared that they have sensed the guidance of our program which verifies the effectiveness of our method. As for question 2, the average score is 3.87, 96.75% of the max score, means that most of our preset aiming-points are been watched. The average score of question3 to 6 is 12, 75% of the max score 16, shows a relatively high satisfaction about the guidance.

6 CONCLUSION AND FUTURE WORK

In this paper, we tried to diminish the gap between the designer and the player with an attentive camera which involved the designer's predefined aesthetic rules to guide the player's behaviors implicitly. Visualized interfaces are provided for both the designer and the player. Besides, a novel NGL is also proposed to help translate the editing rules of the designer to the constraints of the user's behavior. Experiments and a survey show that our method can provide the user with a cozy and stable guidance while roaming in the virtual environment.

However, our method also confronts with some limitations. First, we only consider two elements,best-view and aiming-point, for the designer to specify. Though they do include the position and also the orientation from which the player can capture good camera composition and valuable information, the designer deserves more rights to display their creativities, such as defining special transitions to these aiming-points (pan, tilt, dolly, and crane), using specific stylized camera, etc. Second, the camera speed in our method is steady and controlled by the player, accelerations and decelerations during the camera manipulation can have great psychological effects on the player's behavior and experience, and should be further considered into our program. Third, our NGL here can only record the operations of the designer. To find correlations with animation scripts and drama scripts and implement an automatic translation are a challenging task that should be put on the agenda.

ACKNOWLEDGMENTS

The work is supported by the National Science and Technology Support Program(No.: 2012BAH62F02-02).

REFERENCES

Dan Amerson, Shaun Kime, and R Michael Young. 2005. Real-time cinematic camera control for interactive narratives. In *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology*. ACM, 369–369.

Paolo Burelli. 2012. *Interactive virtual cinematography*. Thesis. IT University Of Copenhagen.

David B. Christianson, Sean E. Anderson, Li-wei He, David H. Salesin, Daniel S. Weld, and Michael F. Cohen. 1996. Declarative camera control for automatic cinematography. AAAI Press, 148–155.

Quentin Galvane, Marc Christie, Chrsitophe Lino, and Remi Ronfard. 2015. Camera-on-rails: automated computation of constrained camera paths. In *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games*. ACM, 151–157.

Michael Gleicher and Andrew Witkin. 1992. Through-the-lens camera control. *ACM SIGGRAPH Computer Graphics* 11, 4 (1992), 331–340.

Stephen Hughes and Michael Lewis. 2000. Attentive camera navigation in virtual environments. In *Systems, Man, and Cybernetics, 2000 IEEE International Conference on*, Vol. 2. IEEE, 967–970.

Arnav Jhala and R. Michael Young. 2005. A discourse planning approach to cinematic camera control for narratives in virtual environments. In *Proceedings of the 20th national conference on Artificial intelligence*, Vol. 1. AAAI Press, 307–312.

Niels Joubert, Dan B Goldman, Floraine Berthouzoz, Mike Roberts, James A Landay, and Pat Hanrahan. 2016. Towards a drone cinematographer: Guiding quadrotor cameras using visual composition principles. *arXiv preprint arXiv:1610.01691* (2016).

C Koch and S Ullman. 1985. Shifts in Selective Visual Attention: Towards the Underlying Neural Circuitry. *Human Neurobiology* 4, 4 (1985), 219–27.

Tsai-Yen Li and Xiang-Yan Xiao. 2005. An interactive camera planning system for automatic cinematographer. In *Proceedings of the 11th International Multimedia Modelling Conference*. IEEE, 310–315.

Patrick Olivier, Nicolas Halper, Jon Pickering, and Pamela Luna. 1999. Visual composition as optimisation. *Artificial & Simulation of Intelligent Behavior Workshop âĂŽ99 Symposium on Ai & Creativity in Entertainment & Visual* 9, 2 (1999), 92–99.

Jan Ondřej, Julien Pettré, Anne-Hélène Olivier, and Stéphane Donikian. 2010. A synthetic-vision based steering approach for crowd simulation. *ACM Transactions on Graphics (TOG)* 29, 4 (2010), 123.

Federico Peinado and Pablo Gervás. 2004. *Transferring game mastering laws to interactive digital storytelling*. Springer, 48–54.

Alejandro Ramirez and Vadim Bulitko. 2015. Automated planning and player modeling for interactive storytelling. *IEEE Transactions on Computational Intelligence and AI in Games* 7, 4 (2015), 375–386.

Craig W. Reynolds. 1999. Steering behaviors for autonomous characters. In *Proceedings of Game developers conference*. 763–782.

Hong Yu and Mark O Riedl. 2012. A sequential recommendation approach for interactive personalized story generation. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*. International Foundation for Autonomous Agents and Multiagent Systems, 71–78.