IADIS International Journal on Computer Science and Information Systems
Vol. 7, No. 2, pp. 165-177
ISSN: 1646-3692

INTERACTIVE VIDEO AND CAMERA POSITION FOR VIRTUAL ENVIRONMENT

David Běhal, *Comenius University, Bratislava, Slovakia*

Jana Dadová, *Comenius University, Bratislava, Slovakia*

ABSTRACT

In this work we address the extended interaction with video in virtual walkthrough. We present an algorithm for automatic extraction of camera positions during the video recording. The algorithm creates a b-spline curve, which represents the camera movement. We propose using this curve as an interactive element. The user can with help of this element change his position in video file according to his position in the virtual walkthrough and not only according to time. With this element we extend the standard interaction with video and also the user has additional information about his position during the presentation. We also present an algorithm for object tracking inside video and creation of object representative. We are using this representatives as an information for the users about additional information for selected object. We extended the interaction within the video and the user can view other information on demand.

KEYWORDS

Camera position, interactive video, virtual museum

1. INTRODUCTION

In these days more and more museums, galleries, but also other companies and institutions use World Wide Web for presenting their exhibitions or work. Most of web presentation consists of certain text, images and also some multimedia elements such as video or audio. Our goal is to provide presentation tool based on video to present cultural heritage, therefore our work is mostly oriented on virtual museums and galleries, but the solution we propose might be used in every virtual presentation.

The simplest way to present a visual cultural or historical object is by using a photo gallery. However, for this purpose, pictures has some disadvantages. Firstly, they are static without any interaction and secondly they have only small angles of view. In one photo mostly one object is captured and only from one point of view. On the other hand, the advantage of

using pictures and photo galleries as a presentation tool is that they do not take a lot of space and bandwidth. This type of presentation is used by many museums [1] or virtual cities [2]. A more advanced way of presentation the exhibitions is to create panorama photos or object panoramas. The visitor can rotate his view or the object itself, so there is also interaction, but it is still an image. We can also find panoramas on the web pages of museum [3] and [4], but we can find this type presentation also in other museums.

On the other hand, we can create 3D models of objects for the presentation in the museum and also create a virtual walkthrough around them. This type of presentation is interactive and visitors can move and rotate the objects. However, it takes a lot of time and effort to create realistic model, because some of the historical objects are very complex. On the other hand we can reconstruct also objects which were destroyed or damaged. Then the visitor gets the idea how the object looked before damage. 3D models as a presentation of museum is used in [5] or [3]. Very good survey of previous techniques on web pages of museums is in [6].

Another way how to present a museum or a gallery is to show a video tour around the exhibition. The author or curator of the exhibition can present its most interesting parts. This type of presentation, which was for the first time used in [7], is dynamic and mostly the visitor can control it by using standard elements like Rewind or Fast forward. Although the visitor can control the walkthrough and jump from one position in video file to another, he is not as free as in virtual 3D space. Majority of the videos has also an audio track, which can be for example some commentary, sound or music.

Our goal is to propose an algorithm to create a virtual video presentation, from recorded video sequences. In this algorithm we have to extract the camera positions from our video sequence, to get the information about camera position for specified time. We represent these positions with approximation curve, which is the representation of camera movement. This curve is used to interactively control the position in the video, not according the time as usual time line do, but according the camera position during the recording.

Second goal of our work is create an algorithm to represent additional information about an object in video with a dot, which will move over the video as the object did in video sequence. This dot is clickable and provide additional information about data. For example if we have an old document in video, we can create a link in video to the digital copy of this document. If the visitor would like to see the document in detail, s/he can click on the dot and s/he will be transferred to the digital copy of this document.

2. BACKGROUND AND RELATED WORK

In this section, we will define important terms for our work and also we will mention work which is related to ours. Also we will discuss mathematical background to better understand our algorithm.

The first term we have to define is “object”, because we use this term to represent already captured object. So it is an object within the video. The “object representation” is an virtual representation of an object, for example bounding box or centre of the object. Last important term is “non-linear video”, which is a set of video files. This video sequences are organized into oriented graph as you can see on fig.1. As the fig. 1 shows we have video sequence for hall, and for rooms 1 to n. At the end of the hall visitor can decide where he will continue his

visit. Therefore the visitor has choice how to continue. If s/he choose the Room 1, the video which represents Room 1 will be played. At the end s/he can decide where to continue now.

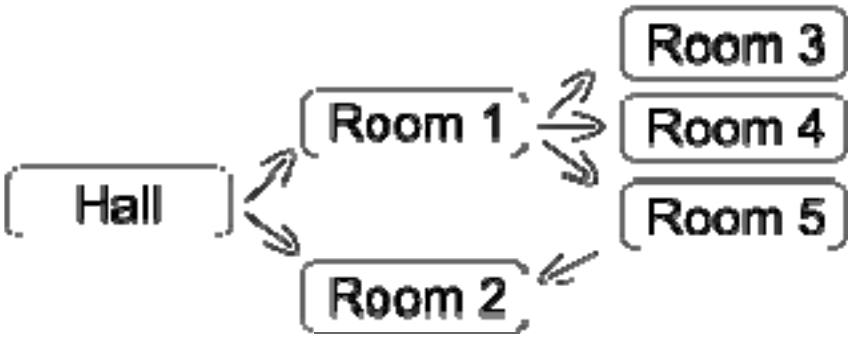

Figure 1. Graph of non-linear video.

2.1 Interaction

Our definition of interaction is based on [8]. From the 21 defined virtual interaction combination is for our case useful just two of them. First the interaction between user and avatar, where we think about user as the visitor of our exhibit and avatar is represented with the graphical user interface. The second interaction is held between user and the objects. In this case we think about interaction between visitor and our object representation.

Basically there are three common interaction elements in standard video players. Firstly, the player has some buttons such as PLAY, PAUSE, STOP and FAST-FORWARD to control the video. Second element is a time line to make a linear jump in video content. This interaction element is linked to time position in video so the user can control the video in 1D space and the only aspect he controls is time. The last common element is a slider to control the volume of the commentary, sound or music. The standard interaction with video does not provide interaction with the content of the video.

There can be found a lot of research in area of interaction and interactive graphics design. Most of the work is in 3D graphics. The authors of [9] presented solution for interaction inside CAVE. In [10] authors presented low cost device to interact with the avatar inside a virtual 3D space. Also companies as [11] or [12] use interaction with video. They add the advertisements inside the video frame and you can find more information about products. In [13] authors from Microsoft presented the Microsoft Kinect as a device to interact with 3D world. We take inspiration from this solution for our video based interaction. But also in 2D world of video and images we can find extended interaction, like in work of Kwiate and Woolner, who have presented story-telling in images and videos of cultural heritage in [14].

2.2 Camera Parameters

As we mentioned before our goal is to extract some data from video sequence. We need to get the camera positions from uncelebrated video sequence. This part of our work is some kind of preprocessing and we will use the research of Marc Pollefeys, mostly the paper [15]. As we

will mention much of the terms from this papers and also terms defined in [16], we will explain some of them also here.

The camera capture process is represented by pinhole camera model. In this model is the camera represented by **M**, 4x3 projective matrix. This matrix represents all the camera parameters, intrinsic and extrinsic. The extrinsic parameters are translation and rotation of camera, so the position and the view vector can be computed. The intrinsic are often written in calibrating matrix. In our work we are focusing on the camera position. However we have not the calibrated camera, so we do not know the calibration matrix. Therefore we will follow the solution from [15].

There are several other researchers, which are solving the problem of structure and motion reconstruction. In work [17] authors added an virtual object inside a video which was not calibrated. In paper [18] authors propose a solution to calibrate camera if there is no real-time condition.

3. PROPOSED ALGORITHM FOR VIDEO PRESENTATION

In this section we offer our algorithm to create a video presentation from a non-linear video. However we have to add some additional constrains on the video sequences. Each video sequence we will process has to be continuous in mean of camera position. Therefore the video should not consist of any cuts fade-outs, fade-ins etc. So the camera movement for one sequence should be continuous curve. In places where the non-linear video is connected the end and the start frame of the next sequence should be the same as much as possible. This condition is only to detect also video connections and it is not necessary. It is possible to make the connections also later in the process manually. We can divide our algorithm to two parts. In first part we are getting the camera positions from the non-linear video. We can call this part as preprocessing, because it will be done just once. In the second part we will create an interactive element from this data, this is done on the fly according the prepared video presentation.

3.1 Preprocessing

The preprocessing part has four main steps. As we mentioned before we have to extract the camera positions from uncalibrated video sequence. When we have our positions, we have to approximate them to get smaller amount of data. If we use them all we can have one point per frame, and if we have 1 minute long video sequence it will be 1500 positions to remember. To eliminate this amount of data we are approximating the position with a b-spline curve, with less control points. Therefore we have not to remember so many points. This have to be done for all sequences.

When we have the curves we can add them into a ground plan of our museum or gallery. This has to be done manually, so the process is only semi automatic. The creator of the video presentation have to scale, rotate and translate each curve to fit the room in the ground plan. In next section we will explain why this has to be done. After all curves has its absolute position inside the background image, we can save them in an XML structure which is representing our video presentation.

3.2 Interaction Element

In this part we are going to draw the ground plan as background and all curves inside a video player we developed in our previous work. Next we are going to use this curves as an clickable interactive element. In this part we have to find the nearest curve to point of click and get the time for that camera position. Here it is important that our curves are parameterized by time. That means the parameter is actual time in video sequence.

When we consider video in context of cultural heritage, we can define 3 types of past. The first "past" is the past from the history point of view. That represents how the events occur chronologically. For example the painter painted his pictures in some order, so one picture is older than another. The second "past" is from creator point of view. As he prepare or capture the video, he put all frames into an chronological order. This means that in the same gallery some pictures can be seen sooner and some later in the video sequence. This order is not so clear in means of non-linear video, where is not defined which is video sequence is next. Therefore we can define "past" from the visitor point of view. This past represents the order in which the visitor saw our non-linear video. This "past" is also used when we jump in time in video sequence. Although the timeline in standard video represents that from t=0 to actual t have been seen, it may not be the truth. For example if we are in time t_0 and we jump to t_1 we have not seen the video between t_0 and t_1. Therefore in our curve we highlight only that parts, which have been seen by the visitor. The skipped frames will be marked as unseen.

3.3 Algorithm

We can form the proposed algorithm into followed steps. As an INPUT for our algorithm we have the non-linear video and the ground plan of the building where the museum is. On the OUTPUT we have virtual video presentation, ready to be published on www. The steps of the algorithm are:

1. choose from the set of non processed video files one video sequence
2. process the video sequence as follows
    (a) find corresponding points between two key frames (these are not the same as key frames during coding process)
    (b) from the corresponding points get the fundamental matrix
    (c) get the projective reconstruction
    (d) with help of auto calibration get the metric reconstruction
    (e) get the extrinsic camera parameters
    (f) get the camera positions and camera rotations for every key frame
    (g) save first 10 and last 10 images of video sequence to get context with other video sequences
3. get approximation curve which approximate the camera positions,
4. find the optimal number of control points
5. save the control points
6. connect the video with the previously processed, if it is possible
7. repeat steps 1-6 until all videos are processed
8. change curves size and position and rotation according the ground plan (user input needed)
9. save the recalculated control points into XML structure

10. load the structure in video player
11. draw the curve and create the interactive element

4. SOLUTION

In this section, we will describe our solution and our implementation of the previous algorithm. As we mention before, our preprocessing work is based on the [15], to extract the camera positions. For implementation purposes we used the OpenCV library [19] and also a libmv library [20], both are open source libraries good for motion and structure reconstruction. The goal of our work is to extend the interaction with video based presentation for museum or gallery. The preprocessing is just necessary step to get the camera positions. Therefore we are not going to improve algorithms we used.

4.1 Preprocessing Implementation

The preprocessing part can be divided into steps, which we are shortly describing. We will do this steps for all video sequences from our non-linear video.

4.1.1 Corresponding Points

First we have to get the relation between each pair of key frames. For this we will use the OpenCV function cvGoodFeaturesToTrack, which uses Shi Tomasi algorithm. Whit this algorithm, we will get points that are good for tracking. Theoretical 7 points should be enough to count fundamental matrix from it, but in most implementation the RANSAC algorithm for counting fundamental matrix is used, therefore we will require more points. Point detection is shown on fig. 2.

Figure 2. Screenshot from point detection step.

When we have our initial set of points, we have to find them in the second key frame. We are using Lucas-Kanade optical flow to track the points over frames. It is implemented in function *cvCalcOpticalFlowPyrLK*, which will give us the new positions of tracked points. Fig. 3 shows the process of point tracking.

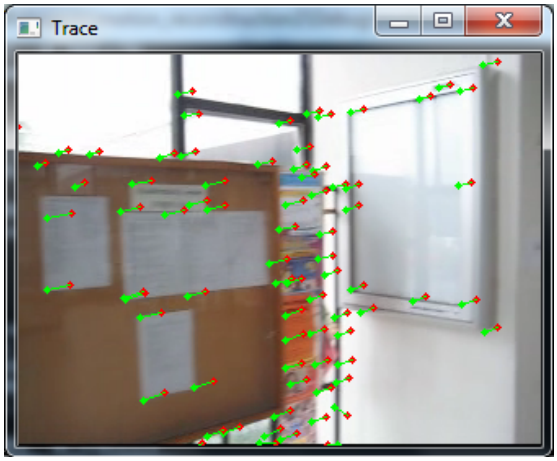

Figure 3. Point tracking during video sequence.

4.1.2 Fundamental Matrix

When we have set of paired points we are calculating the fundamental matrix. For that we use the RANSAC implementation in OpenCV library. If we have some outliers, the algorithm will find them and remove them from sets. Fundamental matrix is a 3x3 matrix with dim (F) = 2. All inliers should satisfy the condition (1) $u^T F u' = 0$.

4.1.3 Initial Frame and Projective Reconstruction

This step is based on [15], therefore we will not discuss it here. To get the 3D point we will use function *cvTriangulatePoints*, which will return the set of 3D points, which are projected to first and second image with projection M_1 and M_2. Next we will add another frames according to [15]. This will cause that we will have projective reconstruction. However we need to get metric reconstruction to get the camera position. We have to use auto calibration as proposed in [15]. This process of auto calibration is implemented in libmv library. From this process we can get the metric reconstruction and therefore the translation and rotation of the camera. The only information we are missing is the scale factor and the position and rotation in the ground plan (starting point). For this missing information we will need the input from the user to scale and place the camera position into the ground plan.

4.1.4 Video Sequence Connection

Next step is to connect the video sequences, if it is possible. This method will help the author of the video presentation to place the curves inside ground plan. We use our stored first and last frames of video sequences to compare if the scene is the same. We count the difference limit in color space and if the difference is below a given threshold then we assume that this two videos are connected. We have to do this on more than 2 images.

4.1.5 Curve Approximation

From previous steps we have camera positions, which we are going to approximate with a cubic b-spline. We use Gauss's method of least squares implemented in library called GeometricTools [21]. We are looking for minimal amount of control points that will approximate the points with minimal error. That means we set an error function that measures the sum of distances between camera positions and the point on the curve for the same time. In the next step we present the results to the user and he have to arrange the curves into the background. After he finish he can save curves into an XML schema in the form of the control points.

4.1.6 XML Schema

As we mentioned before, we store all information for non-linear video in XML file. Now we present an overview of this XML. We have to store the sources of video files, connections between video files and curves for video files. We also store in the XML file the positions and curves for object representation. This XML file is input for the second part, when we will display the result.

4.2 Interaction

The main part of our work is to represent the camera position in sense of curve and use this curve as an interactive element to control the video playback. We call this element *virtual path*. Therefore the user will have two possibilities how to interact with the position in video sequence. First the standard way, where he change the time, which will change the played video. The second more intuitive way according to museums and galleries is to change to video with changing the camera position. The user is doing the same as the author of the video did. He was changing the position of camera.

4.2.1 Curve Interpretation

As first step we have to load our XML structure and interpret *virtual path* as an interactive element. We developed an application called Streamboat, which has been already using interactive elements. The *virtual path* is on fig. 4.

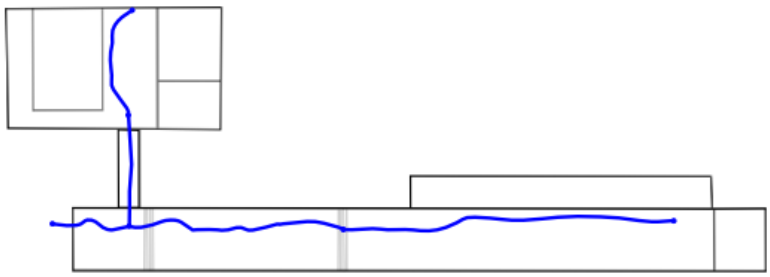

Figure 4. Interactive element *virtual path*.

Each curve is representing one video sequence from non-linear video. Only one video sequence can be active at each time. On fig. 5 is the active curve represented by the blue color,

the green color represents another selectable video sequences and the gray color represents video sequence with missing video source.

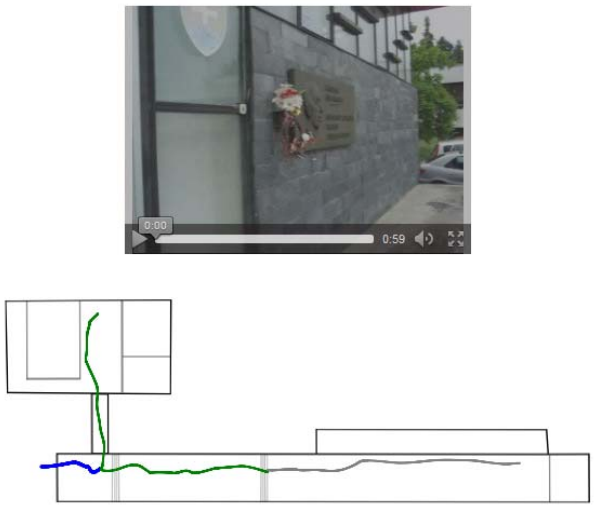

Figure 5. Active video sequence (green), selectable video sequences (blue), not selectable video sequences (gray).

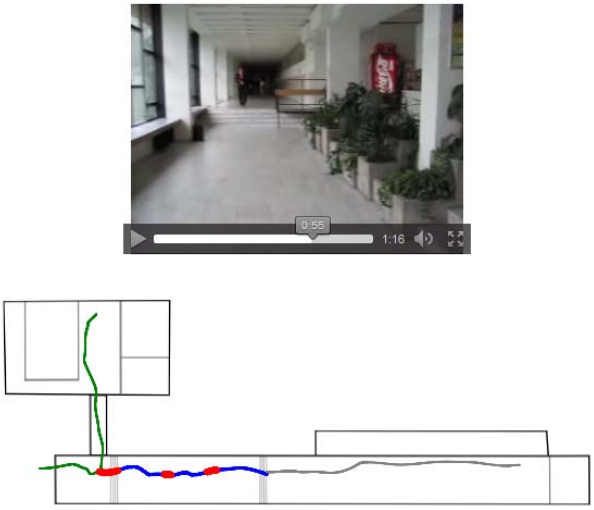

Figure 6. Not visited part in active video (blue), visited part in active video (red).

With the change of the time in video we also draw over the curve positions we have already seen. If we change the time in video let's say from t_0 to t_1, where $t_0 < t_1$, we will not draw curve over the camera positions representing this time segment (t_0, t_1). Fig. 6 represents the highlighted and not highlighted part. Therefore we draw only past from visitor point of view. Parts that has not been visited by user will not be highlighted.

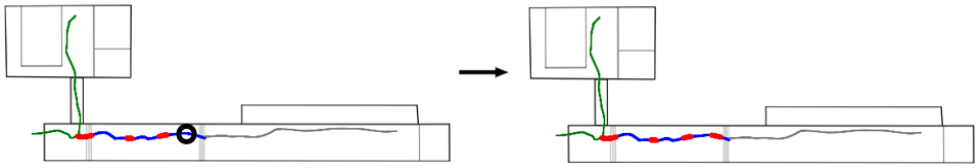

Figure 7. Change of position according camera position (yellow spot).

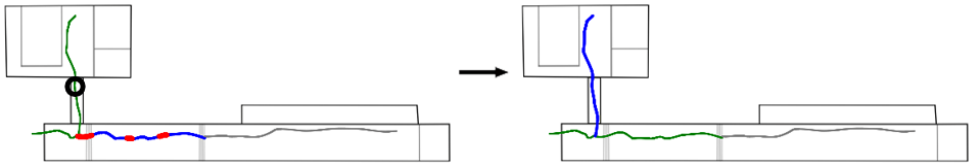

Figure 8. Change video sequence.

The active curve is also clickable, therefore the user can select the position he want to see (fig. 7). The video will change time according the parameter of the selected position. This interaction gives us also the information about actual position inside the museum. If the visitor clicks on not active curve, appropriate video sequence become active. Therefore the visitor can also jump from one room to another (fig. 8). With the non-linear video, where the user can select his next step from more than one video sequences, we presented a novel approach to interaction with video.

4.2.2 Video Sequence Connection

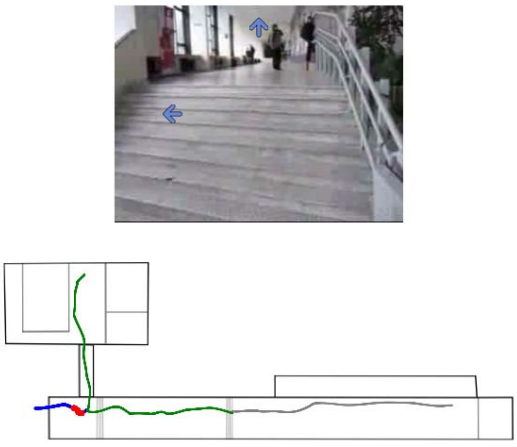

Figure 9. Arrows over video represents connecting video sequences.

At the end of the video sequence we can also display arrows to continue watching the exhibition if there is connected video sequence. For example from the fig 1. At the end of the hall video sequence our video player will display two arrows to continue to room 1 and room 2. Similar example is on fig 9 where is the screenshot from the app.

5. OBJECT INTERACTION

Beside the *virtual path*, we suggest to add some additional information about the objects inside the video. Although in the video we have a sound or commentary, we can still add some more information such as images, links or various texts that will introduce the nature of the object and hence attract the visitor. In work [20] authors propose the solution by adding more information about a picture, and they hide this information inside the image. We suggest to do it another way, because hiding this information in video seems to be much more difficult.

The additional information can be stored in extra files or database and are shown only on visitors demand. For this purpose we will use our object representation, which will be a dot represented the centre of the object. With the help of the algorithms mentioned in preprocessing part, we can interactively select object and track it over the video sequence. However we are just tracking the centre of the object and we approximate this movement with another cubic b-spline.

In the place of object representation we put a dot which is a clickable object and it will represent some additional data (fig. 10). If the visitor clicks on the dot, the content on demand will be shown in the same web page under the video player. Therefore s/he will not lose the context, s/he can still continue watching video.

Figure 10. The brown dot represents that we have some other information about object.

In our solution we have to store just control points of the curve and the start time and end time, when should be the object visible. The additional data are sent only if the visitor requests them. The dot can be also replaced by a bounding rectangle which does not hide the object itself. In some commercial video application we can find this approach to display advertisement over some part of object, which is clickable and you can get information where to buy the product. We transferred this approach also into museum presentation.

6. RESULTS

We have tested our solution on the non-linear video recorded at our faculty. We tested each step separately and also the whole process, which has created a video presentation of the faculty. The preprocessing steps were quite slow, it takes 1 to 2 frames per second. But this is

not a problem, because this part needs to be done, just once. And it is possible to find better algorithms than we used from OpenCV.

The correspondence step and tracking step was accurate. If there were some mistakes, the algorithm from OpenCV mark that points and they were removed from next process. The fundamental matrix computation has problems if the video have been still between two selected key frames. Therefore if the fundamental matrix was not found properly, we selected another key frame, until it was found.

To find the connection between two video sequences we have to use more frames, not just two. If we record the video and in point we can choose two connected videos, we can record just one at a time, therefore the scene is changed in the second one. The process of selecting the control points was very good and the approximation was good also with the small amount of control points. The size of the curve is so small, that in some cases we could get less control points than we get.

We did not find a comparative interactive video presentation for virtual museums and galleries. Therefore we brought a novel approach to cultural heritage video presentation.

7. FUTURE WORK

In future work we are going to finish a usability testing of the interface. We would like to add also the view information to the *virtual path*, so that the user will see in which direction the camera is rotated. In this solution we use just x, y coordinates of the camera movement, in the future work we may represent also z coordinate.

8. CONCLUSION

In our work we have connected the art and cultural heritage with the technology and proposed the solution for the video based presentation with a novel interactive element. This element is used to change the position in video not according time as in usual video, but according the camera position.

We propose an algorithm, which will create a video presentation from uncalibrated non-linear video. The process is semi-automatic, the author have to scale and place the curve over the ground plan of the museum or gallery. It gives the opportunity to explore the museums from the comfort of your home and watch the exhibition presentation.

Our approach is an alternative way for museums and galleries to present their exhibitions. The visitor has less freedom than in 3D virtual walkthroughs, but it still provides more information than images or panoramas.

ACKNOWLEDGEMENT

Our research and paper was partially supported by the Slovak Scientific Grant Agency (VEGA), project No. 1/0602/11.

REFERENCES

[1] Leonardo da Vinci programme. The virtual museum of european roots [online]. http://www.europeanvirtualmuseum.it/, december 2011.

[2] ItalyGuides.it. Virtual rome [online]. http://www.italyguides.it/us/roma/rome_italy_travel.htm, december 2011.

[3] Musée du Louvre. The museum of louvre [online]. http://www.louvre.fr, december 2011.

[4] Virtual Museum of Canada. Virtual museum of canada [online]. http://www.museevirtuel-virtualmuseum.ca/, december 2011.

[5] Andrej Ferko, et al.. Virtual museum technologies [online]. http://www.sccg.sk/~ferko/VU007.pdf, december 2011.

[6] Ivana Varhaníková. Virtuálne bánovce nad bebravou. Master's thesis, Fakulta matematiky, fyziky a informatiky, Univerzita Komenského, Bratislava, 2009.

[7] Peter Clay. Surrogate travel via optical videodisc. Massachusetts Institute of Technology, Dept. of Urban Studies and Planning, 1978.

[8] Lars Qvortrup. Virtual interaction: interaction in virtual inhabited 3D worlds. Springer, 2001.

[9] Michael Wegner and Markus Wacker. Making people move – walking techniques in a cave. In WSCG 2010 Communication papers proceedings, pages 63_75, 2010.

[10] Polceanu Mihai, Popovici Alexandru, and Popovici Dorin-Mircea. A system for panoramic navigation inside a 3d environment. In WSCG 2010 Communication papers proceedings, pages 213_219, 2010.

[11] Microsoft Corporation. Video hyperlink [online]. http://adlab.microsoft.com/Video-Hyperlink/, február 2010.

[12] YouTube LLC. You tube [online]. http://www.youtube.com, december 2011.

[13] Dorin Comaniciu and Peter Meer. Mean shift: a robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(5):603_619, May 2002.

[14] Karol Kwiatek and Martin Woolner. Embedding interactive storytelling within still and video panoramas for cultural heritage sites. In Proceedings of the 2009 15th International Conference on Virtual Systems and Multimedia, VSMM '09, pages 197_202, Washington, DC, USA, 2009. IEEE Computer Society.

[15] Marc Pollefeys, Luc Van Gool, Maarten Vergauwen, Frank Verbiest, Kurt Cornelis, Jan Tops, and Reinhard Koch. Visual modeling with a handheld camera. Int. J. Comput. Vision, 59:207_232, September 2004.

[16] Richard I. Hartley and Andrew Zisserman. Multiple View Geometry in Computer Vision. Cambridge University Press, New York, NY, USA, 2 edition, 2003.

[17] Kurt Cornelis, Marc Pollefeys, Maarten Vergauwen, Luc Van Gool, and K. U. Leuven. Augmented reality using uncalibrated video sequences. In Lecture Notes in Computer Science, page 2001, 2001.

[18] Simon Gibson, Jon Cook, Toby Howard, and Roger Hubbold. Accurate camera calibration for o_-line, video-based augmented reality. In ISMAR, pages 37_46, 2002.

[19] Gary Bradski and Adrian Kaehler. Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly, Cambridge, MA, 2008.

[20] Project libmv. libmv a structure from motion library [online]. http://code.google.com/p/libmv/, december 2011.

[21] LLC Geometric Tools. Geometric tools [online]. http://www.geometrictools.com/Documentation/Documentation.html, december 2011.