Draft submitted to Journal Computers & Graphics, Elsevier. Accepted October 2018.
Online: https://www.sciencedirect.com/science/article/pii/S009784931830178X

A Navigation Paradigm Driven Classification for Video-Based Rendering Techniques

Rafael K. dos Anjos[a,b,*], João Pereira[b,c], José Gaspar[c,d]

[a] Faculdade de Ciências Sociais e Humanas da Universidade Nova de Lisboa
[b] INESC-ID Lisboa
[c] Instituto Superior Técnico, Universidade de Lisboa
[d] Institute for Systems and Robotics (ISR-Lisboa)

Abstract

The use of videos as an input for a rendering process (video-based rendering, VBR) has recently been started to be looked upon with greater interest, and has added many other challenges and also solutions to classical image-based rendering (IBR). Although the general goal of VBR is shared by different applications, approaches widely differ regarding methodology, setup, and data representation. Previous attempts on classifying VBR techniques used external aspects as classification parameters, providing little insight on the inner similarities between works, and not defining clear lines of research. We found that the chosen navigation paradigm for a VBR application is ultimately the deciding factor on several details of a VBR technique. Based on this statement, this article presents the state of art on video-based rendering and its relations and dependencies to the used data representation and image processing techniques. We present a novel taxonomy for VBR applications with the navigation paradigm being the topmost classification attribute, and methodological aspects further down in the hierarchy. Different view generation methodologies, capture baselines and data representations found in the body of work are described, and their relation to the chosen classification scheme is discussed.

Keywords: Video-Based Rendering, Data Representation, Application, Navigation Paradigm, Free Viewpoint Video
2000 MSC: 10.020, 10.080, 20.020, 20.050, 20.090, 110

1. Introduction

For a long time video has been used in our daily lives as the media that more closely recreates an event as we live it in the real world. The recent popularization of personal video cameras and video content distribution has been pushing the scientific community to expand the traditional video format beyond its classical restrictions such as reproduction speed, which gave birth to slow motion videos, and most recently the viewpoint restriction. The process that uses video as input in order to create novel rendered content is generally defined as video-based rendering. This field shares goals and challenges with image-based rendering, while having the extra time dimension that is non-existent in its counterpart. By analyzing the visual content of these images, one tries to extract enough data to add processed information to the existing content or to create novel views that extrapolate the original experience.

Video-based rendering is a topic that combines computer graphics and computer vision; competences from both areas of knowledge are needed. A great effort is made by each community to build the bridge between the two areas. Video-based rendering is without a doubt a challenging field of work.

Different paradigms of user interaction have been proposed for VBR applications, each one allowing users to navigate through the content in a different manner. Thus, creating widely varying lines of work and methodologies to be followed. Each group of applications face different problems, and apply different methodologies and steps on each level of the typical VBR pipeline. Previous classification schemes presented in the past focus either on external aspects to the techniques (e.g. type of input/output, level of automation). These, however, do not present clearly identifiable classes of techniques and methodologies in which one can easily group and classify newly developed work. Moreover, previous state-of-art reviews of VBR works have used a definition that was tied to specific methodologies and data representations, which as research in this field progresses, ceases to be accurate.

This article reviews and classifies VBR works in different groups with the most high level classification parameter being the navigation paradigm, while giving insight on the chosen methodologies, data representation, and techniques in the VBR pipeline. This document will start by defining video-based rendering, the taxonomy to be used in this article, and the VBR pipeline. Followed by a state of the art report on video-based rendering applications and data representation, comparing the most popular trends and grouping similar techniques in general categories. Finally, conclusions and insight will be given on what is the current trend of research, where research should be focused for the near future, and what is there to expect from future work.

*I am the corresponding author
Email address: rkuffner@fcsh.unl.pt (Rafael K. dos Anjos)
URL: https://rafaelkuffner.github.io (Rafael K. dos Anjos)

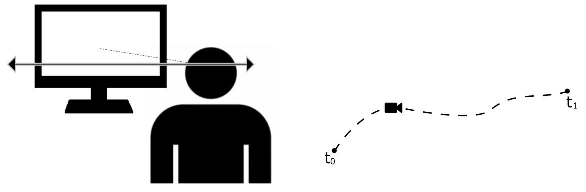
1.1. Video-based rendering definition

Video-based rendering is a term that has been applied to a wide range of techniques, sometimes in a more broad way than it usually is, and other times focused on only a specific type of application. So it is important to establish the definition that will be used on this survey. The term was firstly used on the article by Schödl et al. [1] referring to image-based rendering techniques extrapolated to the temporal domain, using two-dimensional images of a scene to generate a three-dimensional model and render novel views of the scene.

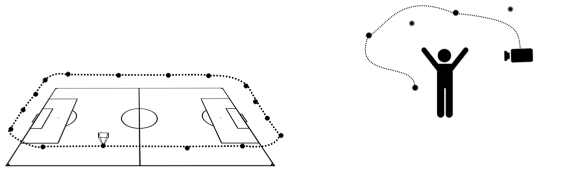
The book from Magnor [2] defines video-based rendering as the process of fusing image-based rendering with motion capture in order to generate a novel view. Borgo et al. [3] on their more broad survey classifies at a top level the techniques under the definition of video-based graphics (a more generalist definition for VBR), focused on creating new content (other videos or 3D reconstructions) based on video input, and video visualization that would encompass the attempts of allowing the user to see video from new/synthetic points of view not previously recorded.

The survey from Stoykova et. al [4] focused only on 3D time-varying reconstruction, more in line with the classical definition of Magnor [2], and would be only a subset of the previous classification, as also Szeliski [5] who stays with the classical definition.

The common ground among all different definitions made at different points in time is the shared goal of creating novel viewpoints of a certain scene, not necessarily sharing a methodology as suggested by Magnor, or a specific type of input, as suggested by Borgo et al. We also consider scenarios where depth information or three-dimensional models are used combined with videos, since the goal of view synthesis is still shared. Considering this, we define VBR as **the process generating novel views of a recorded event on video.**



(a) Head-face parallax (b) Navigation through time



(c) Navigation through viewpoints (d) Free virtual camera

Figure 1: Different user interaction paradigms for VBR, which are the basis for our classification.

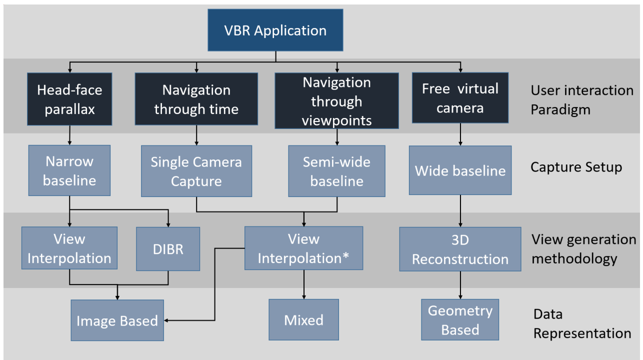


Figure 2: Diagram showing the used classification scheme for this survey. User interaction paradigm defines what capture setup is needed, which relates closely to view generation methodologies and data representations.

1.2. Navigation paradigm driven classification

The chosen definition accommodates a large group of works which have considerable differences among them. Not only different devices are used for input, but also processing techniques, and type of data representation will differ considerably from one work to another. Due to this fact, defining clear groups of applications considering every applied technique is not viable. Few attempts of classifying VBR techniques as a whole have been made, with surveys commonly focusing on classifying each type of application or lower level techniques.

Authors have classified techniques according to taxonomies based on external aspects of the application such as level of automation, type of output and input information [3], or had to focus on a more specific domain of applications where classification is simpler [4].

We found that the chosen navigation paradigm for a VBR application is ultimately the deciding factor on three key aspects of a VBR technique: **View generation methodology, capture setup, and data representation.** The amount of freedom that is given to the user, and the type of navigation through the novel views (which we refer to as the navigation paradigm), guides the decisions made regarding how to capture and generate the content. Four navigation paradigms were found in the reviewed literature. 1)Head-face parallax, where the user can navigate a plane parallel to the visualization plane. (Figure 1a), 2)Navigation through time, where the novel views are generated in a fixed timeline that the user can control (Figure 1b), 3)Navigation through viewpoints, where one is allowed to navigate between predefined viewpoints (Figure 1c), and 4)Free Virtual camera, where there are no positional restrictions to navigation (Figure 1d). These will be analyzed in depth in section 6.

Figure 2 shows the choices for each one of these aspects according to the user interaction paradigm of the application. By classifying the techniques according to the five possible combinations of choices that can be made, we have clear different classes of works that one can easily identify and apply to different real world problems. Each one of the described aspects and grouping of applications will be described in Section 2.

2. Video-based rendering applications

As stated in Section 1.1, the main objective of VBR applications is the generation of novel views. We selected sixty-one articles from over the last 15 years which share this objective, yet use different approaches. We sought to answer a group of questions for each one of them:

1. What capture device was used?
2. Which lower level techniques were applied?
3. Which higher level techniques were applied?
4. What view generation methodology was used?
5. What was the data representation used for that application?
6. What was the capture setup used?
7. What is the navigation paradigm applied to it?

Questions 1-3 give us insight on individual decisions each one of the works make, but did not reveal clear groups of applications, or informed us about high level methodologies. This is due to the fact that these decisions are relatively low level, and techniques are applied with different purposes and in different combinations, not necessarily defining an approach or application.

Questions 4-6 are higher level decisions which clearly relate to each other and allow us to classify different works into categories. Methodologies for view generation (4) were identified in our review, which have strong relations to other of the raised questions, and also allow different types of application for each one of them. Data representation (5) will decide what data is stored, and what can be generated in these novel views. Finally, the capture setup (6) is directly related to the navigation paradigm (7) of the application, since it decides the spatial limits of the interaction. We considered these four aspects to be the most relevant on defining a VBR application. Nevertheless the navigation paradigm (7) was found to be the key deciding factor on what approach is used, as discussed in section 6.

We will start by describing the different capture devices used in the reviewed VBR works, since this decision transversely influences methodology, setup and representation. The following sections (4 5 and 6) will describe the answers to the last four questions listed before, giving insight on each one of the reviewed works, explaining its relevancy in a VBR application and the navigation paradigm in hand (Section 6).

3. Data capture

The data capture step in a VBR process will define what type of input information we have available for all the following steps. The presence of either depth information, a 3D coordinate system or skeletal information, will directly affect the used data representation, and also influence the chosen view generation methodology. Also, different types of devices are better suited for specific types of setup baselines and navigation paradigms. This aspect will be discussed in Section 6

Besides conventional color cameras, color-depth, laser scanners, and mixed inputs have been used on VBR and IBR applications.

3.1. Color cameras

Main efforts in image and video-based applications are focused on capturing images with conventional color cameras [6] [7] [8] [9] [10], not only due to the lower cost of the devices, but the popularity of the developed methodologies (code publicly available) and the amount of data already available that could be used for applications such as shown in the work of Ballan et al. [11]. Although being a bigger challenge than using more complex and informative data, it is of great interest to be able to use raw images for a VBR process, specially from mainstream media outlets.

Regarding the type of cameras suited for the VBR, there are some core requirements that should be met for the data to be useful. Data acquisition using all the cameras must be possible to trigger from an external switch so the several different sources can be synchronized, and the camera should be able to record in progressive scan mode, not interlaced half-images [2] which should be common in modern cameras. Camera resolution and recording speed are up to the application objectives, not being a general and essential parameter as the previous two. Other useful features are the ability to record raw pixel data, in order not to deal with images preprocessed by the camera internal hardware, flexible to high f-stop numbers, high dynamic range, good color properties, and other features. The book from Magnor [2] gives useful insight on some of the issues that should be considered for both image and video-based rendering.

3.2. Color depth cameras

Another input device that has been recently popularized on VBR applications is the color depth camera. It enables depth estimation to be performed reliably with a single device. Asus Xtion Pro, ZED, Intel RealSense, and most popularly The Microsoft Kinect Sensor have been used due to their real time nature and low-cost. Depth sensors were already an option on the past [12] but recently they were made more accessible and complete with other built-in functions, such as body tracking, which can be used as secondary information in some VBR scenarios. Differently from traditional laser scanners, these devices try to operate in real time, making them suited for VBR, unlike traditional scanners [13] [14] [15] which deliver high quality results, but have long capture times.

Different depth estimation techniques have been used in the commercialized devices. Infrared disparity matching [16] was used in the first Kinect sensor, where a pattern is projected to the scene and recognized by a infrared camera so the distance between recognizable features can be estimated. This was substituted by time-of-flight laser scanning in the newest sensor which has considerably higher precision. Both approaches are not set back by textureless regions as image-based stereo methods [17], but might suffer from interference from sunlight in outdoor scenarios. The ZED sensor uses stereo matching between two color cameras, which combined with spatial localization of the sensor, is able to reconstruct the environment at a higher distance, but lower precision. This approach suffers from lighting variations and low-fidelity reconstruction at textureless regions. Lightfield cameras, or plenoptic cameras have also been recently made commercially accessible, and applied to VBR in different contexts. These devices are essentially composed by an array of micro-lenses and sensors, and allow one to obtain precise information about the captured scene including depth [18]. A strong comparison can be made between them, and a grid disposition of cameras [19, 20], and they have both been used in similar VBR scenarios.

3.3. Hybrid input

Duan et al. [21] showed that is possible to perform fusion between depth maps from stereo cameras and Kinect sensors in real time, having an overall better result than using a single device. The work from Goesele et al. [22] is an example of another type mixed input that combines the raw images with an estimated bounding box for the object to be scanned. Also Ballan et. al [11] take other information as input such as available 3D models for a prior reconstruction of the scenery and better positioning of the cameras, since the input videos are not calibrated by default. The 3D model input does not always guarantee a better result, but having an initial geometry estimate does improve with the efficiency of the technique, as shown by the image-based rendering review from Shum and Kang [23].

4. Novel view generation method

Having captured an event from one or more viewpoints, unrecorded visualizations can be generated through different processes. The chosen methodology will depend on the available data (3D information, images, depth values, etc) and the desired navigation paradigm (navigate freely vs. recorded viewpoints).

Older definitions of VBR mentioned on Section 1.1 defined VBR through the used methodology. Schödl et. al [1] and Magnor [2] defined it as processes that necessarily required reconstruction. The fact that the field evolved in different directions and newer processes and applications were created, we decided to use a definition based on goal only, and use the methodology as one of the classification parameters of a certain work.

4.1. 3D Reconstruction and rendering

The classical definition of VBR was grounded on 3D reconstruction and rendering procedures to generate views [1] since this resembled the traditional process to generate novel views in Computer Graphics (CG). Rendering 3D models into 2D photorealistic images accordingly to the position and orientation of a virtual camera is a straightforward task that has been well documented and investigated by the CG community. When 3D information about the scene is available, any desired viewpoint can be rendered through this process. The outline of this process can be seen in Figure 3

In the VBR context, the 3D reconstruction step poses a challenge because the initial input of the process does not commonly provide three-dimensional information. The inclusion of the recent depth sensors in the capture process could fix the problem but as mentioned in Section 3, using such sensors is not always viable, so we must still consider 3D reconstruction without direct 3D information from the input video streams.

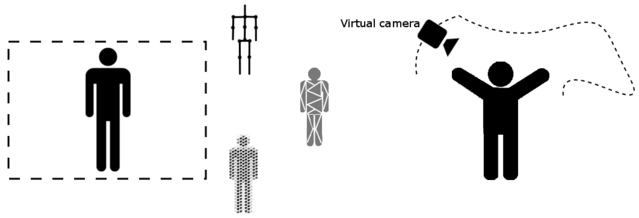


Figure 3: Outline of the 3D Reconstruction and rendering view generation method. Captured data is used to create different types of representations (3D Reconstruction), which are then used differently to create a 3D visualization (rendering).

As we are going to see next, despite of different approaches to provide 3D information for performing the 3D reconstruction, the novel view creation is accomplished by executing afterwards the classical rendering process with the available 3D models or structures that were estimated from the input.

When the focus of the application are human performers (e.g. sports and dance applications), very simplistic 3D information such as an estimated skeleton can be sufficient for novel view generation. Players are segmented from the background, and their skeletons are recognized from the poses captured in video. On the works of Gall et al. [24] and Li et al. [25], a mesh is estimated using a visual hull for the performer so it can be applied to the tracked skeleton. Stoll et al. [26] and Wu et al. [27] move this task to a pre-processing step where depth sensors are used to create an animated model of the performer. The drawback is that changes in the outfit or hair of the performer will not be supported.

Germann et al.[28] has a similar but unique approach, where the same process for estimating the skeleton is used, but instead of applying a 3D mesh to it, segmented billboards of each body part of the performer are applied to the tracked skeleton, this approach is not a pure 3D reconstruction case since the applied textures are view interpolated. We chose to describe it here due to the similarities to the previous approaches.

Volino et al. [29] and Imber et al.[30] use a initial capture of the performer to construct a texture map, which will be applied to the estimated visual hulls in each frame. A skeleton is not estimated on these works, instead a sequence of visual hulls is calculated.

Finally, the most straightforward approach to 3D reconstruction relies on directly estimating depth information from camera inputs, or depth sensors, creating complex three-dimensional structures that will be used for rendering. Zeng et al. [31] and Kuster et al. [32] use directly the input from the Microsoft Kinect for that task. Google Tango [33] and the work from Liu et al. [34] use multiview stereo to estimate depth information, and on the latter, a visual hull is used to define the limits of the human performer that is being captured, refining the MVS process.

The most recent work using this methodology was from Pagés et al. [35], which uses different sources of information

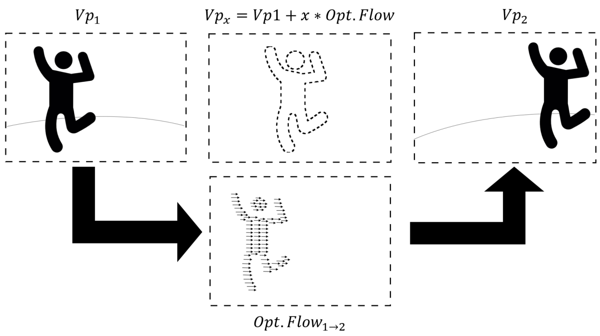


Figure 4: Outline of the View interpolation method. Optical flow between adjacent viewpoints is estimated, and interpolation is performed to create an intermediate point of view.

to create a full high quality 3D reconstruction of a recorded scene. Multiview stereo is used to estimate rough 3D coordinates of each pixel, which is combined with silhouette and skeletal detection to refine the performers mesh. The advantage over similar work [24, 25] is that there is no pre-processing step to estimate a mesh, as it is performed in real time. This allows deformable tissues and hair to be correctly reconstructed.

4.2. View interpolation

When the required novel views are close to a previously recorded video stream, 3D reconstruction step may not be necessary to perform the rendering operation. Chen and Williams [36] described this process on their pioneer work. This approach introduced in 1993 allowed very complex scenes to be rendered through this process, since it is not reliant on the complexity of the objects to be rendered. Szeliski presents this methodology in his survey [5] and also in his own research as one of the basic building blocks for VBR applications.

The scene is captured with an array of aligned cameras, and the relative position between pixels from different viewpoints is estimated through the optical flow from one point to another. These vectors are stored in a ”morph map”, a disparity matrix, which will be used to interpolate the values between each one of the viewpoints and generate the new images on the unrecorded viewpoints, as seen on Figure 4. If the changes are parallel to the viewing plane, the interpolated result is perfect. Also, as mentioned before, the closest the images are to the original viewpoints, the better the estimated results.

One relevant reference is the work from Kanade [37] about the coverage of the Super Bowl XXXV, where the broadcasting team, instead of individual users, was able to cycle seamlessly through the several cameras in the stadium to give more insightful replays. View interpolation and a rough reconstruction which is possible due to the playing field being known, are used to create transition frames between cameras. A similar recent product by Vizrt [38] has been extending the functionality to allow not only transition between cameras but also to generate other points of view. This and similar approaches that combine traditional view interpolation with specialized information have been referred to as ”view interpolation*” in Figure 2.

Goorts et al. [6] uses a similar methodology, but uses multiview-stereo to estimate depths for each point, and render better interpolated images. Similarly, Taguchi et al.[20] , Wang et al. [39], and specially Tanimoto et al. [40] have used MVS, but in order to represent the scene in the Ray-space using the plenoptic function. This representation allows an easier generation of views given the accurate estimation of this space. Ng et al. [41] uses the same methodology but with a more object focused approach, improving the results in object boundary regions. Tanimoto et al. [40] introduced specialized devices to quickly create such representation for small scale object Similarly, recent work from Domanski et al. [42] uses this approach the chosen view generation technique when neighboring cameras are placed in an arc, not in a line (where DIBR is used). For synthetic views that are not the originally captured, an audio interpolation technique is also discussed.

One particular interpolation use case is video stitching, where closely captured sequences are used to generate a wider video. Image-stitching is a classical problem of computer vision and has been widely discussed by the community [43]. When adding a temporal dimension camera stabilization, new challenges have to be considered in the performed interpolation. Efficiency [44], color correction [45], wider baselines [46], ghosting artifacts [47], video stabilization [48] among others. These have been the main focus points in recent research, with each different algorithm and proposed technique being more suited to different type of content. Regarding our VBR definition, they can be considered borderline VBR, as most of the times no completely novel views are being created, but through distortion and interpolation of part of the data, views with wider fovs are generated.

One interesting view interpolation work that must be mentioned is the one from Ballan et al. [11] which applies this methodology for a different purpose: to navigate between casual uncalibrated captures of the same performance. A rough three-dimensional reconstruction of the background is performed using SfM to estimate each camera position. Then view interpolation is used to create transition frames between one viewpoint to the other. The performer is represented as a billboard naturally changes during transitions, and the background information is interpolated between viewpoints. This work extends the work from Kanade [37] and [38] to a more casual scenario, where the capture is performed in an uncontrolled scenario. Similarly, Lipski et al.[49] presented a similar approach where the user could navigate in time and space on interpolated views of neighboring videos.

4.2.1. Temporal interpolation

Interpolation within a single input video has been used in different VBR works as a methodology for generating novel content. The two main techniques in this category found in the revised papers are the hyperlapse, and video summarization. These methods have been referred to as ”View Interpolation*” in Figure 2.

The hyperlapse appeared as an adaptation of the time-lapse videos to scenarios where the camera is moving. Time-lapse videos will typically record one frame every x seconds and combine everything in a single video. If the camera is moving during the video capture process, it will generate unstable videos that are unsuitable for watching. Hyperlapses will try to temporally stabilize such videos.

The groundbreaking work of Kopf et al. [50] uses SfM to create a rough reconstruction from the environment based on different frames. A stable path is calculated through the 3D estimate of the environment, and new frames are rendered through that path at the new camera positions. Using interpolation between different frames, texture information is projected to the extrapolated proxy geometry, creating novel views of unrecorded data, based on existing frames.

The work from Joshi et al [51] uses purely image information to create a hyperlapse. By dropping frames that destabilize the camera flow, a smooth video is created. In this particular work, new information is not created by any interpolation method, putting it in the border line between VBR and video editing. A similar approach by Halperin et al .[52] also selects the best frames, but creates novel information by using such dropped frames to increase the field of view of the recorded video, creating unrecorded information for visualization. A similar case is the work of Lai et al. [53] which does hyperlapses for 360 degrees videos, creating a smooth path for the camera by focusing on certain points of interest throughout the video. While no unrecorded information is created, all the resulting frames are created through an automatic process, and the camera path is created through interpolation between different positions of subsequent 360 degrees frames.

Video summarization is another area where temporal interpolation is applied and also has borderline VBR work. Different methods have been applied where frames are also selected in order to only keep only the most relevant information. DeMenthon et al. [54] does it through curve simplification, while the work of Ma et al. [55] uses a user attention model to detect which instants in the video are relevant and should be visualized as a whole, and which can be summarized. While a new video is created, no unrecorded information and visual information is produced.

On the other hand, the "Video Summagator" from Nguyen et al. [56] can create complete novel views while summarizing the video. It uses the complete video information to create a 3D representation of the video as a whole. The authors demonstrate scenarios where a panning camera could be used to stitch a wider background through temporal interpolation, so foreground elements could be visualized over a complete overview of the camera's trajectory.

One notable video stitching example that can be placed slightly off the curve, and more in the line of other VBR applications is the work from Agarwala et al. [57], where a single moving video is used to create a panoramic texture. Both time and content are manipulated to transform a sweeping motion of a camera into a wider video, manipulating the content in each different time frame to match the past, and create a seamless animation.

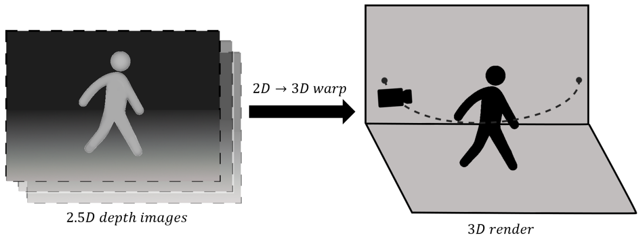


Figure 5: Depth Image Based Rendering. A set of 2.5D depth images is warped to create a 3D render that can be visualized from a set of positions.

4.3. Depth image-based rendering

Depth image-based rendering as a view generation methodology has been acquiring popularity in the recent years since depth data is easier to be captured or estimated with modern cameras or specified sensors. In their quality assessment work on FVV [58] Sandić-Stanković et al. consider DIBR to be the main view generation methodology applied in the field. Novel views are rendered through warping the Color Depth data into three-dimensional information, which then can be viewed through chosen viewpoints. This process, shortly named "3D warping", was introduced by McMillan [59] in his 1997 work, and is summarized in Figure 5. The work from Zitnick et. al [60] can be considered one of the recent precursors of this line of research. In this work depth is estimated through MVS and used for DIBR. The resulting dancers data set has been used as a standard benchmark in the majority of work described below.

The novel view generation methodology is the same but each group of works has focused on different aspects of the process.

Yoon et. al [61] and Muller et. al [62] have presented specific data representation for this field (5), focusing on compression of data. This line has been followed by several authors [63] [64] [65] [66] al [67] and will be discussed in Section 5.

Due to the fact that the estimated depth values might not create a complete scene due to occlusions, or depth discontinuities might exist due to differences in estimation from one viewpoint to the other, other works have focused on in-painting and hole filling. Zhu and Li's approach [68] performed hole filling through background segmentation where missing information can be recovered from other views or frames, only interpolating between neighboring pixels when necessary. Rahaman and Paul more recent work introduces Gaussian mixture models so when the 3D warped views are created and holes are filled with information from a different perspective, boundaries are less perceivable due to low correspondence between the views. Daribo and Saito [63] and Yang et. al [69] techniques have worked towards this goal using different data representations, and fitting the hole filling task in the process of representing the data.

With the recent advances in rendering capabilities of mobile devices, several works have been published in adapting DIBR to mobile platforms. The work from Shi et al. [70] from 2009 talks about rendering data in a remote location, and just request the result, since processing power on the device could be not enough for interactive view synthesis. Miao et al. [71] has a different approach to minimize interaction delay, since the transmission might be slower than the generating the view locally. Their approach performs local rendering that is halted in case the result comes through the network. Most recently, Malia and Debono [72] divide frames into smaller tiles so they can be processed in different threads, since more recent devices have better processors.

Huszák [73] focused on less bandwidth use. His work describes a specialized network structure for FVV where each node can who both render and cache rendered views, so view synthesis results can be re-used by other clients, since they are stored in the network nodes. Li et al.[74] proposed a standard for LTE networks, which reduces the bandwidth in 30%, optimizing the routing of the transmitted data.

5. Data representation

After going through the lower steps of the VBR pipeline, information about the captured scene is encoded in a suitable format for the chosen rendering process. We found three groups of representation in the surveyed works. Geometry based representations, where the scene was modeled as a group of three-dimensional objects along the time. Mixed representations, where part of the scene is modeled through images, and part through geometry. And image based representations, where the scene is stored in bi-dimensional matrices with color and optionally depth.

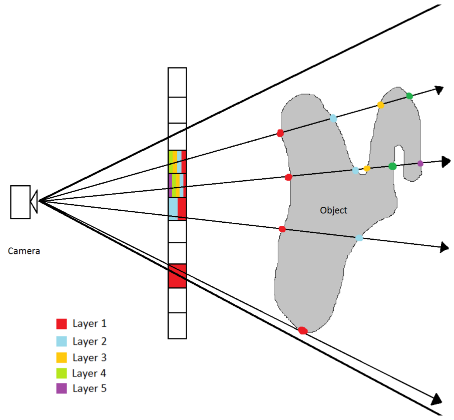
Although it might usually be considered just an implementation detail, the data representation on a VBR process is tightly related to the chosen methodology for novel view generation, and also to the desired type of application. Different representations enable the development of alternative methodologies for view generation. As seen on Section 4, reconstruction was performed in different ways, all creating different types of data.

5.1. Geometry-based representation

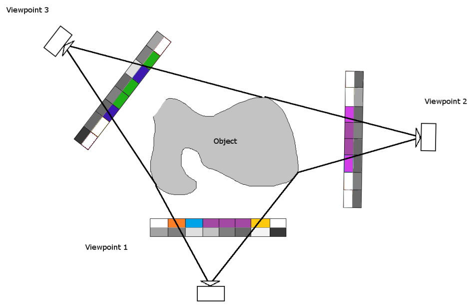
The most straightforward way to represent a scene is through geometric primitives. It has been the go-to approach in most rendering scenarios. On VBR, they result from a 3D reconstruction process, and employed in traditional rendering to generate novel views.

Animated meshes were used in several works [24] [26] [25] [27] where the target of the visualization is one or more human performers which can be segmented properly in order to estimate the skeletons. Scenarios with large groups, occlusions, and close interactions pose challenging issues. When a static mesh has been captured in a previous step for that single performer [26], this representation is very efficient. The recent work from Pagés et al. [35] performs 3D mesh reconstruction using different sources of information (MVS, pose information, visual hull), creating complex geometric information.

When a skeleton can not be reliably tracked, surfaces [34] [32], point clouds [33] and octrees [31] can be used. These are classically used for static reconstructions, but can be applied in dynamic scenarios. Although more flexible and being complete representations (contain full and precise information about the objects in the scene), they are less efficient for VBR. Applying temporal compression requires specialized algorithms [75], while image-based representations can apply video compression, which is always evolving. Geometry-based representations are usually applied in real-time applications where storage and compression is not an issue.



(a) Layered Depth Image



(b) Multiview plus Depth

Figure 6: Two different image-based representations based on depth.

5.2. Image-based representation

Image-based representations are independent of scene complexity, being well suited for these scenarios. On the other hand, they are typically discrete, and do not allow certain rendering effects that require precise geometric information. Specifically for VBR, they have the advantage of enabling ordinary video compression techniques to be applied to them, which is not possible with geometry-based representations.

On several view interpolation scenarios [37] [5] [20], ordinary video streams for each recorded viewpoint are the only information about the scene in hand. Although effective, further work has shown that depth information is important not only for view-interpolation and DIBR when pursuing accurate results. Color plus depth video streams have been used for this matter [60] [69], where depth information is estimated through MVS or captured with specialized sensors.

Two other image-based representations have been presented as alternatives to RGBD streams. Multiview plus depth (MVD) by Merkle et. al [62] , and the Layered depth video (LDV) by Yoon et. al[61].

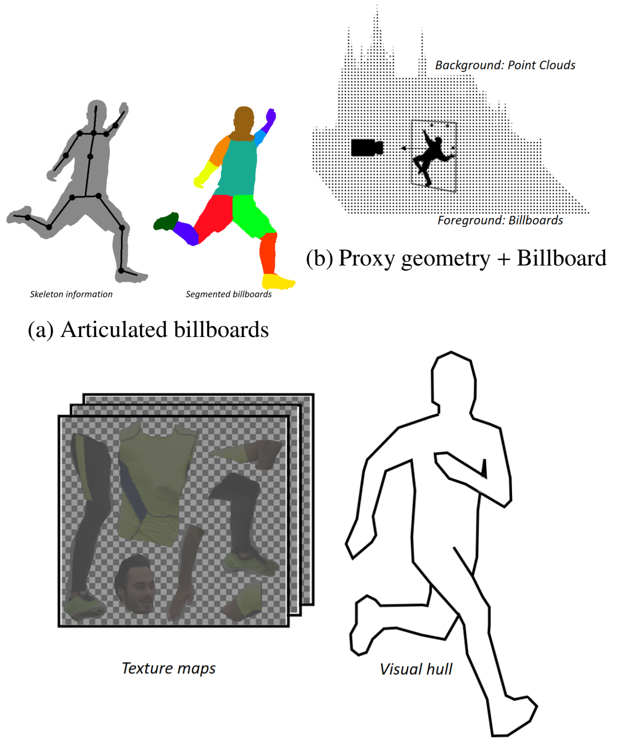
Multiview plus depth coding is performed by encoding each separate RGBD viewpoint as a different stream, and compression is applied separately to each stream as illustrated on figure 6b. The Layered depth image format introduced by Shade et. al [76] (the basis for LDV) is one of the most efficient on rendering 3-D objects with complex geometries. It represents a scene as viewed from a chosen point of view, but storing not only color values for each pixel but also depth information and other features that can be used for the rendering process. One of their key characteristics is the fact that they store more than one point for each pixel. Figure 6a shows an example of an LDI formed from a three-dimensional object. Rays are emanated from a certain viewpoint and intersections with an object are stored with depth and color information. When the same ray goes through more than one point of the object, the subsequent intersections are added to the back layers. Typically the front layers are more populated, with only residual information on the last layers.

Layered depth videos [61] extend this representation to a video format, and their authors argue that is a more efficient than the multiview plus depth approach on this type of setups. Both use this image-based representation to apply video compression algorithms to the stream, with the difference of the former (MVD) keeping every stream separate, and using them to generate a new viewpoint on the viewer side, and the latter (LDV) warping the scene to a single point of view, eliminating some redundancy, but possibly losing some information due to thresholding. One recently introduced alternative was the Multiview layered depth image (MVLDI) [77], which applies a similar process than the LDI one, but uses a global thresholding approach, not image-based. Also, each layer is encoded according to a different viewpoint. By doing this, the advantages of the LDI can be extended to wider baseline scenarios and more flexible navigation paradigms.

Finally, Plenoptic videos [40] [39] have been successfully employed on view-interpolation. They capture color and depth information from different viewpoints and represent it as the Plenoptic function (7D) [78], or the Lumigraph [79], its 4D simplification. With θ and ϕ being the azimuth and elevation angle of the rays, and λ the wavelength, it is calculated at a position (v_x, V_y, V_z) in space, and on the VBR scenario, the function is 7D due to the time component. So we have the following form to the function, which can be considered a complete scene description:

$$p = P(\theta, \phi, \lambda, V_x, V_y, V_z, t) \tag{1}$$

Although it is a complete scene description, on a real scenario we cannot capture the scene from every possible viewpoint. In practice, data is captured with a narrow grid with several cameras, or cameras based on arrays of micro-lenses (plenoptic or light field cameras). This representation is used by sampling this function at the eye positions (v_x, V_y, V_z) representing the capture viewpoints, and interpolating the values given by each one of them to generate intermediate views. Such representation is promising for 3D television, but is still far from being accessible for research.



(a) Articulated billboards

(b) Proxy geometry + Billboard

(c) Visual hull + Texture-maps

Figure 7: Mixed representations with part represented by a geometric reconstruction, and part by sequences of images.

5.3. *Mixed representation*

Although MVD and LDI contain geometric information in the form of depth values, we still consider them as image-based representations due to the fact that they are stored as images, and warping needs to be performed during rendering to obtain the three-dimensional values. Examples in this category are partly represented by sequences of images, and partly by geometry.

As mentioned in section 4.1 Germann et. al [28] uses articulated billboards (Figure 7a) . Skeleton information (geometry) is stored alongside images which are interpolated and applied to each skeleton. Also the approaches from Volino et al. [29] and Imber et al.[30], which use a simplified mesh through a visual hull (geometry) combined with sequences of textures (images) that are mapped into it (Figure 7c). Ballan et. al [11] has a similar approach but keeping the background geometry static since it is only used to track positions of each viewpoint in order to generate the transitions (Figure 7b).

Finally Ng et. al [41] use the Plenoptic function representation, but segmented to individual objects in the scene, which can be considered a mixed representation, due to the fact that individual objects in the scene are separated from each other, making the representation more tied to the content of the scene than other image-based representations.

All of these representations aim to combine advantages from both worlds. Having three-dimensional representation allow one to generate novel viewpoints further away from the original recording points, and using image-based representations, data compression is considerably easier to be applied, and the representation complexity is scene independent. It is important to notice though, that in all of the reviewed works, strong assumptions about the storing content needed to be made. Typically these were used to represent human performers in controlled conditions, such as a studio capture setup, or a sports event where the layout and the captured elements are known.

6. Baseline of the data acquisition setup, and navigation paradigm

Multi input setups are the typical scenario for VBR. Only a small portion of the surveyed works have used a single input camera, and could be classified as VBR. Devices can be placed in a narrow, wide, or semi wide-baseline setup as seen on Figure 8. In a narrow set up, the cameras are placed closer to each other with little disparity between adjacent views, usually with each device parallel to each other. A wide setup typically aims to capture a scene or object from all different perspectives, having the cameras placed further away from each other, where disparity between views is now desired, not avoided. The semi-wide scenario would be a step in between where disparity is avoided but different viewpoints are desired.

On multi-streams approaches there is also the need of extrinsic calibration for the cameras, i.e. know the relative positions between them. In controlled environments this can be done by using markers detected by the camera [80] [9], but on dynamic environments the most common approach is to track features using structure from motion [11, 81, 82], providing a reliable position calibration for the camera. When depth cameras are used, specific systems that take advantage of higher level information have been proposed, as in the work of Sousa et al. [83] where skeleton information is used to quickly calibrate a group of Kinect sensors, and point cloud information is used to fine-tune the resulting calibration. A parallel problem to this is the stream synchronization problem, which can be solved by an external centralized trigger on controlled scenarios [9] [6]. Audio stream aligned can be used on uncontrolled scenarios [21] [11].

Although the general goal of VBR is the same across applications, each one of them have different specific goals depending on the desired navigation paradigm, as seen on Figure 1. On all reviewed works, we found that navigation paradigm is tightly connected to the capture setup. According to the objective of the application, the setup will be adapted, and all other factors mentioned previously are then a consequence of this decision. Due to this fact, this section groups each work by the camera setup, and explain the typical application for each setup, and how it relates to the previously raised questions.

6.1. Narrow baseline applications: Head-face parallax

One navigation paradigm associated to a free viewpoint videos consists of a moving user in front of a screen while having the perception of depth through parallax. By adjusting the viewpoint to the position of the user's eyes, this effect is possible. Since the user performs movements in a parallel plane to the captured scene, novel views only need to be generated in this domain. For this purpose, a narrow capture setup parallel to the captured scenario will suffice for the desired results. Figure 9 summarizes this application group.

When a narrow capture setup is used, cameras and/or depth sensors are arranged in a line [60] or in a grid [20, 19], according to the freedom of choice of views provided by the application. Here we also consider lightfield capture and plenoptic cameras. A close comparison can be made between them and a grid narrow-baseline disposition, as mentioned in Section 3, and they have been successfully used to generate novel views in a head-face parallax scenario [84]. This setup is ideal for a performance type of recording, where the audience is supposed to be facing a stage from a certain direction.

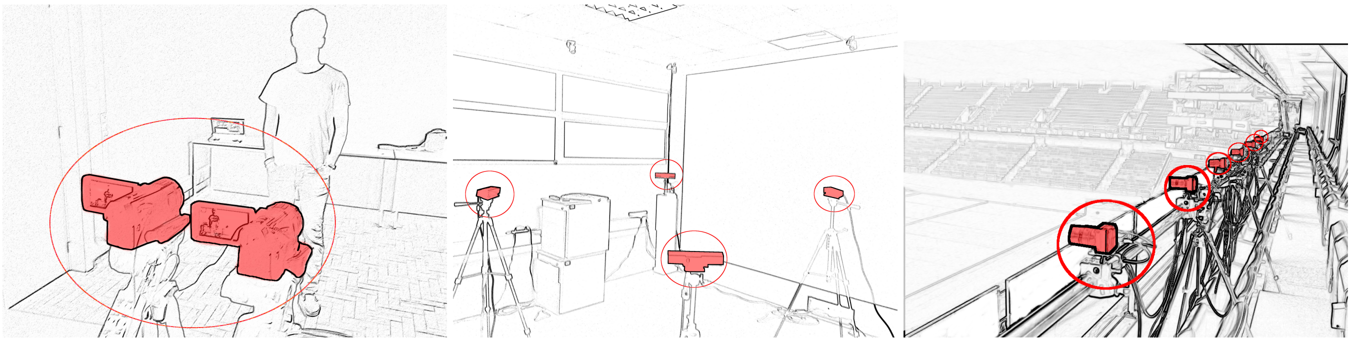
Methodologies such as view interpolation (VI) and DIBR have good performance in this scenario due to the small disparity between adjacent viewpoints. Applications that perform video stitching also fall in this category, where the user either visualizes the whole stitched video, or has a head-face parallax experience. 3D reconstruction will create incomplete results, since only one side of the object is being captured. VI has been used when depth estimation is not reliable enough for rendering, but used sometimes as an aid to the interpolation process.It was also applied when lightfield reconstruction is performed, as mentioned in Section 4.2. DIBR have been used in all other works reviewed in this survey.

All strategies for this setup have used image-based representations because they are meant to work on any kind of data with no expected restrictions, and as mentioned previously, image-based representations are independent of the complexity of the scene.

6.2. Semi-wide baseline applications: Navigation through viewpoints

A small subset of works reviewed in this survey aims a similar experience to wider setups, where the user can navigate in a full circle around a scene, but the content of the visualization is more complex than having a single performer. Similarly to wide setups with mixed representations, strong assumptions can be made about the content, but the type of result desired is closer to narrow baseline applications. Either navigating through camera viewpoints, or generating intermediate viewpoints but not widely far from the defined grid of visualization. For this sense, a "less narrow", or "semi-wide" setup is used (Figure 10).

Instead of performing 3D reconstruction with view interpolation in some components such as the work from Volino et. al with articulated billboards [29], the preferred approach is view interpolation supported by three-dimensional information about the scene (marked with a * in Figure 2). On sports scenarios [37] [11] [38], this information has been used to generate transition frames between viewpoints. Given the fact that the reconstruction is rough, the user never gets to properly visualize intermediate frames. The remaining works in this category [41] [6] create intermediate viewpoints, but use background geometry information to support this view generation process.



(a) Narrow setup with color cameras for stereo matching

(b) Wide baseline of a low-cost setup for a 360 degrees capture of a subject.

(c) Semi-Wide baseline setup for Sports

Figure 8: Different capturing setups for VBR with different input devices.

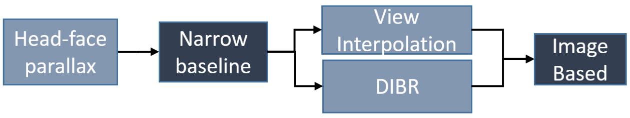


Figure 9: Head-face parallax application classes

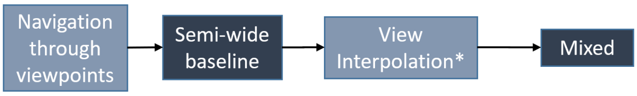


Figure 10: Navigation through viewpoints applications

6.3. *Single camera capture applications: Navigation through time*

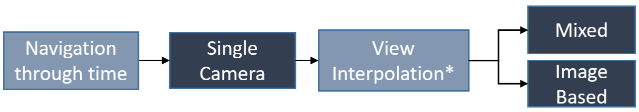


Figure 11: Navigation through time applications

In the works where only a single viewpoint is captured, novel views can only be generated by temporal interpolation, extrapolating the data captured by that single viewpoint [50, 56]. In this case, the user experience is similar to watching a normal video, albeit seeing novel rendered images or modified perspectives. Some of the works in this category are difficult to compare to other VBR works, due to the fact that a true novel view is sometimes not created, but merely chosen from a group of available views. Also due to the fact that the navigation paradigm does not change much from a traditional video. However, since novel content is created and such works are traditionally considered to be VBR works, we include them in our classification as their own category (Figure 11).

The data representation applied in these works is typically image-based, with certain works [50, 53] using it to estimate a proxy geometry, making their data representation mixed. Mixed representation will typically support more complex systems which is able to generate more novel content.

6.4. *Wide baseline applications: Free virtual camera*

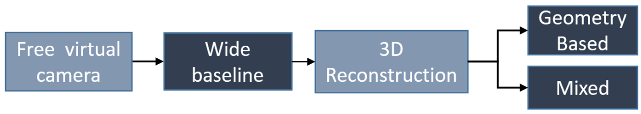


Figure 12: Free-camera navigation applications

When the created application aims to generate novel views all around the subject of visualization, and not only on a parallel plane in front of it, a wide setup must be used (Figure 12). Interaction with the video is usually done indirectly, moving a virtual camera freely around the point of interest.

This type of setup has been used on scenarios where the focus of the video are human performers in a controlled environment [24] [26] [25] [85].

A wide-baseline setup can be comparable to a single depth sensor moving widely around a scene for static reconstruction purposes [34] [33], since the camera will end up assuming positions equivalent to a wide-baseline setup.

Because the viewpoint disparity is too high for view interpolation and DIBR, 3D reconstruction was the methodology applied in all of the surveyed works. Regarding data representation, when stronger assumptions about the content of the scenes could be made such as in sports scenarios, or controlled environments, mixed representations could be used [28] [29] [30]. All the remaining papers in this category used different geometry-based representations. Figure 9 shows the different choices that can be made in this application group.

7. Conclusion and future trends

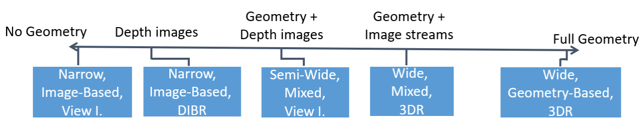


Figure 13: Classes of applications (setup, representation, view generation methodology) placed on a straight line according to the similarities between their approaches regarding to geometry used in their data representation.

As explained in Section 1.2 and seen on Figure 2 the different works can be separated in a hierarchy according to the aspects reviewed above. Each navigation paradigm is closely tied to a camera setup, and to one or two methodologies or data representations. These two aspects are chosen according to the type of data to be captured.

Summarizing the reviewed aspects, DIBR and View interpolation have been used in uncontrolled scenarios, where image-based representations can be applied. When strong assumptions can be made about the scene in hand, mixed representations have been used for view interpolation or 3D reconstruction. Geometry based representations have been applied on generic scenarios with low requirements regarding quantity of data, or when the subject of the free viewpoint video was a human performer in a controlled environment. Finally, view interpolation in the form of timely interpolation has been used primarily for single camera setups.

The presented classification for VBR groups different approaches not only into clearly identifiable classes that share methodologies and problems, but also gives meaningful insight on how they operate on the traditional VBR pipeline. Figure 13 organizes the reviewed classes in a straight line according to similarity between each approach.

With our navigation paradigm driven taxonomy, four different classes which have their own line of research were identified. Despite of the fact that they share similar techniques, each one aims to solve different application requirements. We have noticed that geometrical information, including depth values, plays an increasingly important role in the three classes. This is justified by the hardware advances, namely, more powerful graphic cards and low-cost depth sensors availability. Approaches such as view-interpolation were initially a solution to complex scenes in VBR since full geometry could not be processed in real time to generate views. We believe 3D reconstruction will increase even more their relevance in this field as a methodology.

DIBR has been a good example of an approach that integrates well the geometric component because it is able to apply image-based representations which can be easily compressed in the temporal domain for transmission. With the continuously increasing requirements regarding viewing resolution, these aspects will become more significant. Successful data representations for future VBR applications have to include compression mechanisms, as has been seen in the growing body of work which adapts DIBR to mobile phones and networks.

Acknowledgments

This work was supported by the European Research Council under the project (Ref. 336200). This work was also supported by national funds through FCT - Fundação para a Ciência e Tecnologia, with reference UID/CEC/50021/2013.

References

[1] Schödl, A., Szeliski, R., Salesin, D.H., Essa, I.. Video textures. In: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH '00; New York, NY, USA: ACM Press/Addison-Wesley Publishing Co. ISBN 1-58113-208-5; 2000, p. 489–498. doi:10.1145/344779.345012. URL `http://dx.doi.org/10.1145/344779.345012`.

[2] Magnor, M.. Video-based rendering. Ak Peters Series; A K Peters; 2005. ISBN 9781568812441. URL `http://books.google.com.br/books?id=RbWzOCocpbMC`.

[3] Borgo, R., Chen, M., 1Daubney, B., Grundy, E., Heidemann, G., HÃ¶ferlin, B., et al. State of the art report on video-based graphics and video visualization. Computer Graphics Forum 2012;31(8):2450–2477. doi:10.1111/j.1467-8659.2012.03158.x. URL `http://dx.doi.org/10.1111/j.1467-8659.2012.03158.x`.

[4] Stoykova, E., Alatan, A., Benzie, P., Grammalidis, N., Malassiotis, S., Ostermann, J., et al. 3-d time-varying scene capture technologies, a survey. Circuits and Systems for Video Technology, IEEE Transactions on 2007;17(11):1568–1586. doi:10.1109/TCSVT.2007.909975.

[5] Szeliski, R.. Video-based rendering. In: 2nd European Conference on Visual Media Production. The Institution of Electrical Engineers, Savoy Place, London, UK; 2005, p. 1–8.

[6] Goorts, P., Ancuti, C., Dumont, M., Rogmans, S., Bekaert, P.. Real-time video-based view interpolation of soccer events using depth-selective plane sweeping. In: Proceedings of the Eight International Conference on Computer Vision Theory and Applications. ISBN 978-989-8565-48-8; 2013,.

[7] Hauswiesner, S., Straka, M., Reitmayr, G.. Coherent image-based rendering of real-world objects. In: Symposium on Interactive 3D Graphics and Games. I3D '11; New York, NY, USA: ACM. ISBN 978-1-4503-0565-5; 2011, p. 183–190. doi:10.1145/1944745.1944776. URL `http://doi.acm.org/10.1145/1944745.1944776`.

[8] Furukawa, Y., Ponce, J.. Accurate, dense, and robust multiview stereopsis. Pattern Analysis and Machine Intelligence, IEEE Transactions on 2010;32(8):1362 –1376. doi:10.1109/TPAMI.2009.161.

[9] Carranza, J., Theobalt, C., Magnor, M.A., Seidel, H.P.. Free-viewpoint video of human actors. ACM Trans Graph 2003;22(3):569–577. doi:10.1145/882262.882309. URL `http://doi.acm.org/10.1145/882262.882309`.

[10] Vogiatzis, G., Hernández, C.. Video-based, real-time multi-view stereo. Image and Vision Computing 2011;29(7):434 – 441. doi: 10.1016/j.imavis.2011.01.006.

[11] Ballan, L., Brostow, G.J., Puwein, J., Pollefeys, M.. Unstructured video-based rendering: interactive exploration of casually captured videos. ACM Trans Graph 2010;29:87:1–87:11. doi:http://doi.acm.org/10.1145/1778765.1778824. URL `http://doi.acm.org/10.1145/1778765.1778824`.

[12] Curless, B., Levoy, M.. A volumetric method for building complex models from range images. In: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques. SIGGRAPH '96; New York, NY, USA: ACM. ISBN 0-89791-746-4; 1996, p. 303–312. doi:10.1145/237170.237269. URL `http://doi.acm.org/10.1145/237170.237269`.

[13] Koutsoudis, A., Vidmar, B., Ioannakis, G., Arnaoutoglou, F., Pavlidis, G., Chamzas, C.. Multi-image 3d reconstruction data evaluation. Journal of Cultural Heritage 2014;15(1):73 – 79. doi: http://dx.doi.org/10.1016/j.culher.2012.12.003.

[14] Huang, H., Brenner, C., Sester, M.. A generative statistical approach to automatic 3d building roof reconstruction from laser scanning data. {ISPRS} Journal of Photogrammetry and Remote Sensing 2013;79(0):29 – 43. doi:http://dx.doi.org/10.1016/j.isprsjprs.2013.02.004.

[15] Besl, P.. Active, optical range imaging sensors. Machine Vision and Applications 1988;1(2):127–152. doi:10.1007/BF01212277. URL `http://dx.doi.org/10.1007/BF01212277`.

[16] Arieli, Y., Freedman, B., Machline, M., Shpunt, A.. Depth mapping using projected patterns. 2012. US Patent 8,150,142.

[17] Lee, S., Ho, Y.. Real-time stereo view generation using kinect depth camera. APSIPA ASC 2011;:1–4.

[18] Bishop, T.E., Favaro, P.. Plenoptic depth estimation from multiple aliased views. In: 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops. 2009, p. 1622–1629. doi: 10.1109/ICCVW.2009.5457420.

[19] Zhang, C., Chen, T.. A self-reconfigurable camera array. In: ACM SIGGRAPH 2004 Sketches. ACM; 2004, p. 151.

[20] Taguchi, Y., Takahashi, K., Naemura, T.. Real-time all-in-focus video-

based rendering using a network camera array. In: 2008 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video. 2008, p. 241–244. doi:10.1109/3DTV.2008.4547853.

[21] Duan, Y., Pei, M., Wang, Y.. Probabilistic depth map fusion of kinect and stereo in real-time. In: Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on. 2012, p. 2317–2322. doi:10.1109/ROBIO.2012.6491315.

[22] Goesele, M., Curless, B., Seitz, S.. Multi-view stereo revisited. In: Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on; vol. 2. 2006, p. 2402–2409. doi:10.1109/CVPR.2006.199.

[23] Shum, H., Kang, S.B.. Review of image-based rendering techniques. In: VCIP. 2000, p. 2–13.

[24] Gall, J., Stoll, C., de Aguiar, E., Theobalt, C., Rosenhahn, B., Seidel, H.P.. Motion capture using joint skeleton tracking and surface estimation. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. 2009, p. 1746–1753. doi:10.1109/CVPR.2009.5206755.

[25] Li, G., Wu, C., Stoll, C., Liu, Y., Varanasi, K., Dai, Q., et al. Capturing relightable human performances under general uncontrolled illumination. Computer Graphics Forum 2013;32(2pt3):275–284. doi:10.1111/cgf.12047. URL `http://dx.doi.org/10.1111/cgf.12047`.

[26] Stoll, C., Gall, J., de Aguiar, E., Thrun, S., Theobalt, C.. Video-based reconstruction of animatable human characters. ACM Trans Graph 2010;29(6):139:1–139:10. doi:10.1145/1882261.1866161. URL `http://doi.acm.org/10.1145/1882261.1866161`.

[27] Wu, C., Stoll, C., Valgaerts, L., Theobalt, C.. On-set performance capture of multiple actors with a stereo camera. ACM Trans Graph 2013;32(6):161:1–161:11. doi:10.1145/2508363.2508418. URL `http://doi.acm.org/10.1145/2508363.2508418`.

[28] Germann, M., Hornung, A., Keiser, R., Ziegler, R., Würmlin, S., Gross, M.. Articulated billboards for video-based rendering. Computer Graphics Forum 2010;29(2):585–594. doi:10.1111/j.1467-8659.2009.01628.x. URL `http://dx.doi.org/10.1111/j.1467-8659.2009.01628.x`.

[29] Volino, M., Hilton, A.. Layered view-dependent texture maps. In: Proceedings of the 10th European Conference on Visual Media Production. CVMP '13; New York, NY, USA: ACM. ISBN 978-1-4503-2589-9; 2013, p. 16:1–16:8. doi:10.1145/2534008.2534022. URL `http://doi.acm.org/10.1145/2534008.2534022`.

[30] Imber, J., Volino, M., Guillemaut, J.Y., Fenney, S., Hilton, A.. Free-viewpoint video rendering for mobile devices. In: Proceedings of the 6th International Conference on Computer Vision / Computer Graphics Collaboration Techniques and Applications. MIRAGE '13; New York, NY, USA: ACM. ISBN 978-1-4503-2023-8; 2013, p. 11:1–11:8. doi:10.1145/2466715.2466726. URL `http://doi.acm.org/10.1145/2466715.2466726`.

[31] Zeng, M., Zhao, F., Zheng, J., Liu, X.. Octree-based fusion for realtime 3d reconstruction. Graphical Models 2013;75(3):126 – 136. doi:http://dx.doi.org/10.1016/j.gmod.2012.09.002. Computational Visual Media Conference 2012.

[32] Kuster, C., Bazin, J.C., Öztireli, C., Deng, T., Martin, T., Popa, T., et al. Spatio-temporal geometry fusion for multiple hybrid cameras using moving least squares surfaces. Computer Graphics Forum 2014;33(2):1–10. doi:10.1111/cgf.12285. URL `http://dx.doi.org/10.1111/cgf.12285`.

[33] Google, . Project tango. https://www.google.com/atap/projecttango/project; 2014.

[34] Liu, Y., Dai, Q., Xu, W.. A point-cloud-based multiview stereo algorithm for free-viewpoint video. Visualization and Computer Graphics, IEEE Transactions on 2010;16(3):407–418. doi:10.1109/TVCG.2009.88.

[35] Pagés, R., Amplianitis, K., Monaghan, D., Ondřej, J., Smolić, A.. Affordable content creation for free-viewpoint video and vr/ar applications. Journal of Visual Communication and Image Representation 2018;53:192 – 201. doi:https://doi.org/10.1016/j.jvcir.2018.03.012.

[36] Chen, S.E., Williams, L.. View interpolation for image synthesis. In: Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH '93; New York, NY, USA: ACM. ISBN 0-89791-601-8; 1993, p. 279–288. doi:10.1145/166117.166153. URL `http://doi.acm.org/10.1145/166117.166153`.

[37] Kanade, T.. Carnegie mellon goes to the super bowl. http://www.ri.cmu.edu/events/sb35/tksuperbowl.html.; 2001.

[38] Libero, V.. Viz libero: Sports broadcasting redefined. http://www.vizrt.com/products; 2014.

[39] Wang, C., Chan, S.C., Ho, C.H., Liu, A.L., Shum, H.Y.. A real-time image-based rendering and compression system with kinect depth camera. In: 2014 19th International Conference on Digital Signal Processing. 2014, p. 626–630. doi:10.1109/ICDSP.2014.6900740.

[40] Tanimoto, M.. Ftv: Free-viewpoint television. Signal Processing: Image Communication 2012;27(6):555 – 570. doi: http://dx.doi.org/10.1016/j.image.2012.02.016. URL v.

[41] Ng, K., Chan, S., Wu, Q., Shum, H.. Object-based coding for plenoptic videos. IEEE Transactions on Circuits and Systems for Video Technology 2010;:2-s2.0-77951122418doi:10.1109/TCSVT.2010.2041820. URL `http://hdl.handle.net/10722/128744`.

[42] Domański, M., Bartkowiak, M., Dziembowski, A., Grajek, T., Grzelka, A., Łuczak, A., et al. New results in free-viewpoint television systems for horizontal virtual navigation. In: 2016 IEEE International Conference on Multimedia and Expo (ICME). 2016, p. 1–6. doi:10.1109/ICME.2016.7552993.

[43] Adel, E., Elmogy, M., Elbakry, H.. Image stitching based on feature extraction techniques: a survey. International Journal of Computer Applications (0975-8887) Volume 2014;.

[44] Hu, J., Zhang, D.Q., Yu, H., Chen, C.W.. Discontinuous seam cutting for enhanced video stitching. In: 2015 IEEE International Conference on Multimedia and Expo (ICME). 2015, p. 1–6. doi:10.1109/ICME.2015.7177506.

[45] Xu, W., Mulligan, J.. Performance evaluation of color correction approaches for automatic multi-view image and video stitching. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2010, p. 263–270. doi:10.1109/CVPR.2010.5540202.

[46] Li, J., Xu, W., Zhang, J., Zhang, M., Wang, Z., Li, X.. Efficient video stitching based on fast structure deformation. IEEE Transactions on Cybernetics 2015;45(12):2707–2719. doi:10.1109/TCYB.2014.2381774.

[47] Jiang, W., Gu, J.. Video stitching with spatial-temporal content-preserving warping. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). 2015, p. 42–48. doi:10.1109/CVPRW.2015.7301374.

[48] Guo, H., Liu, S., He, T., Zhu, S., Zeng, B., Gabbouj, M.. Joint video stitching and stabilization from moving cameras. IEEE Transactions on Image Processing 2016;25(11):5491–5503. doi:10.1109/TIP.2016.2607419.

[49] Lipski, C., Linz, C., Berger, K., Sellent, A., Magnor, M.. Virtual video camera: Image-based viewpoint navigation through space and time. Computer Graphics Forum 2010;29(8):2555–2568. doi:10.1111/j.1467-8659.2010.01824.x.

[50] Kopf, J., Cohen, M.F., Szeliski, R.. First-person hyper-lapse videos. ACM Trans Graph 2014;33(4):78:1–78:10. doi:10.1145/2601097.2601195. URL `http://doi.acm.org/10.1145/2601097.2601195`.

[51] Joshi, N., Kienzle, W., Toelle, M., Uyttendaele, M., Cohen, M.F.. Real-time hyperlapse creation via optimal frame selection. ACM Trans Graph 2015;34(4):63:1–63:9. doi:10.1145/2766954. URL `http://doi.acm.org/10.1145/2766954`.

[52] Halperin, T., Poleg, Y., Arora, C., Peleg, S.. Egosampling: Wide view hyperlapse from egocentric videos. IEEE Transactions on Circuits and Systems for Video Technology 2017;:1–1doi:10.1109/TCSVT.2017.2651051.

[53] Lai, W., Huang, Y., Joshi, N., Buehler, C., Yang, M., Kang, S.B.. Semantic-driven generation of hyperlapse from 360 degree video. CoRR 2017;abs/1703.10798. `1703.10798`; URL `http://arxiv.org/abs/1703.10798`.

[54] DeMenthon, D., Kobla, V., Doermann, D.. Video summarization by curve simplification. In: Proceedings of the Sixth ACM International Conference on Multimedia. MULTIMEDIA '98; New York, NY, USA: ACM. ISBN 0-201-30990-4; 1998, p. 211–218. doi:10.1145/290747.290773. URL `http://doi.acm.org/10.1145/290747.290773`.

[55] Ma, Y.F., Lu, L., Zhang, H.J., Li, M.. A user attention model for video summarization. In: Proceedings of the Tenth ACM International Conference on Multimedia. MULTIMEDIA '02; New York, NY, USA: ACM. ISBN 1-58113-620-X; 2002, p. 533–542. doi:10.1145/641007.641116. URL `http://doi.acm.org/10.1145/641007.641116`.

[56] Nguyen, C., Niu, Y., Liu, F.. Video summagator: An interface for video summarization and navigation. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. CHI '12; New York, NY, USA: ACM. ISBN 978-1-4503-1015-4; 2012, p. 647–650. doi:10.1145/2207676.2207767. URL `http://doi.acm.org/10.1145/2207676.2207767`.

[57] Agarwala, A., Zheng, K.C., Pal, C., Agrawala, M., Cohen, M., Curless, B., et al. Panoramic video textures. In: ACM SIGGRAPH 2005 Papers. SIGGRAPH '05; New York, NY, USA: ACM; 2005, p. 821–827. doi:10.1145/1186822.1073268. URL `http://doi.acm.org/10.1145/1186822.1073268`.

[58] Sandić-Stanković, D., Battisti, F., Kukolj, D., Callet, P.L., Carli, M.. Free viewpoint video quality assessment based on morphological multiscale metrics. In: 2016 Eighth International Conference on Quality of Multimedia Experience (QoMEX). 2016, p. 1–6. doi:10.1109/QoMEX.2016.7498949.

[59] McMillan, L.. An image-based approach to three-dimensional computer graphics. Ph.D. thesis; Citeseer; 1997.

[60] Zitnick, C.L., Kang, S.B., Uyttendaele, M., Winder, S., Szeliski, R.. High-quality video view interpolation using a layered representation. In: ACM SIGGRAPH 2004 Papers. SIGGRAPH '04; New York, NY, USA: ACM; 2004, p. 600–608. doi:10.1145/1186562.1015766. URL `http://doi.acm.org/10.1145/1186562.1015766`.

[61] Yoon, S.U., Lee, E.K., Kim, S.Y., Ho, Y.S.. A framework for multi-view video coding using layered depth images. In: Advances in Multimedia Information Processing-PCM 2005. Springer; 2005, p. 431–442.

[62] Merkle, P., Smolic, A., Muller, K., Wiegand, T.. Multi-view video plus depth representation and coding. In: Image Processing, 2007. ICIP 2007. IEEE International Conference on; vol. 1. 2007, p. I – 201–I – 204. doi:10.1109/ICIP.2007.4378926.

[63] Daribo, I., Saito, H.. A novel inpainting-based layered depth video for 3dtv. Broadcasting, IEEE Transactions on 2011;57(2):533–541. doi:10.1109/TBC.2011.2125110.

[64] Yoon, S.U., Lee, E.K., Kim, S.Y., Ho, Y.S.. A framework for representation and processing of multi-view video using the concept of layered depth image. The Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology 2007;46(2-3):87–102.

[65] Kirshanthan, S., Lajanugen, L., Panagoda, P., Wijesinghe, L., De Silva, D., Pasqual, A.. Layered depth image based hevc multi-view codec. In: Bebis, G., Boyle, R., Parvin, B., Koracin, D., McMahan, R., Jerald, J., et al., editors. Advances in Visual Computing; vol. 8888 of *Lecture Notes in Computer Science*. Springer International Publishing. ISBN 978-3-319-14363-7; 2014, p. 376–385.

[66] Kim, W.S., Ortega, A., Lai, P., Tian, D.. Depth map coding optimization using rendered view distortion for 3d video coding. IEEE Transactions on Image Processing 2015;24(11):3534–3545. doi:10.1109/TIP.2015.2447737.

[67] Merkle, P., Müller, K., Marpe, D., Wiegand, T.. Depth intra coding for 3d video based on geometric primitives. IEEE Transactions on Circuits and Systems for Video Technology 2016;26(3):570–582. doi:10.1109/TCSVT.2015.2407791.

[68] Zhu, C., Li, S.. Depth image based view synthesis: New insights and perspectives on hole generation and filling. IEEE Transactions on Broadcasting 2016;62(1):82–93. doi:10.1109/TBC.2015.2475697.

[69] Yang, X., Liu, J., Sun, J., Li, X., Liu, W., Gao, Y.. Dibr based view synthesis for free-viewpoint television. In: 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), 2011. 2011, p. 1–4. doi:10.1109/3DTV.2011.5877165.

[70] Shi, S., Jeon, W.J., Nahrstedt, K., Campbell, R.H.. Real-time remote rendering of 3d video for mobile devices. In: Proceedings of the 17th ACM International Conference on Multimedia. MM '09; New York, NY, USA: ACM. ISBN 978-1-60558-608-3; 2009, p. 391–400. doi:10.1145/1631272.1631326. URL `http://doi.acm.org/10.1145/1631272.1631326`.

[71] Miao, D., Zhu, W., Luo, C., Chen, C.W.. Resource allocation for cloud-based free viewpoint video rendering for mobile phones. In: Proceedings of the 19th ACM International Conference on Multimedia. MM '11; New York, NY, USA: ACM. ISBN 978-1-4503-0616-4; 2011, p. 1237–1240. doi:10.1145/2072298.2071983. URL `http://doi.acm.org/10.1145/2072298.2071983`.

[72] Mallia, M., Debono, C.J.. Rendering of free-viewpoint video on the cloud. In: IEEE EUROCON 2017 -17th International Conference on Smart Technologies. 2017, p. 9–14. doi:10.1109/EUROCON.2017.8011069.

[73] Huszák, Á.. Advanced free viewpoint video streaming techniques. Multimedia Tools and Applications 2017;76(1):373–396. doi:10.1007/s11042-015-3048-9. URL `https://doi.org/10.1007/s11042-015-3048-9`.

[74] Lee, J.T., Yang, D.N., Chen, Y.C., Liao, W.. Efficient multi-view 3d video multicast with depth-image-based rendering in lte-advanced networks with carrier aggregation. IEEE Transactions on Mobile Computing 2018;17(1):85–98. doi:10.1109/TMC.2017.2707416.

[75] Slomp, M., Kawasaki, H., Furukawa, R., Sagawa, R.. Temporal octrees for compressing dynamic point cloud streams. In: 2014 2nd International Conference on 3D Vision; vol. 2. 2014, p. 49–56. doi:10.1109/3DV.2014.79.

[76] Shade, J., Gortler, S., He, L.w., Szeliski, R.. Layered depth images. In: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH '98; New York, NY, USA: ACM. ISBN 0-89791-999-8; 1998, p. 231–242. doi:10.1145/280814.280882. URL `http://doi.acm.org/10.1145/280814.280882`.

[77] Anjos, R.d., Pereira, J.M., Gaspar, J.A., Fernandes, C.. Multiview layered depth image. Journal of WSCG 2017;25(2):115–122.

[78] McMillan, L., Bishop, G.. Plenoptic modeling: An image-based rendering system. In: Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH '95; New York, NY, USA: ACM. ISBN 0-89791-701-4; 1995, p. 39–46. doi:10.1145/218380.218398. URL `http://doi.acm.org/10.1145/218380.218398`.

[79] Gortler, S.J., Grzeszczuk, R., Szeliski, R., Cohen, M.F.. The lumigraph. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH '96; New York, NY, USA: ACM. ISBN 0-89791-746-4; 1996, p. 43–54. doi:10.1145/237170.237200. URL `http://doi.acm.org/10.1145/237170.237200`.

[80] Sturm, P., Maybank, S.. On plane-based camera calibration: A general algorithm, singularities, applications. In: Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.; vol. 1. 1999, p. –437 Vol. 1. doi:10.1109/CVPR.1999.786974.

[81] Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., et al. Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera. In: Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology. UIST '11; New York, NY, USA: ACM. ISBN 978-1-4503-0716-1; 2011, p. 559–568. doi:10.1145/2047196.2047270. URL `http://doi.acm.org/10.1145/2047196.2047270`.

[82] Newcombe, R.A., Fox, D., Seitz, S.M.. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2015,.

[83] Sousa, M., Mendes, D., Anjos, R.K.D., Medeiros, D., Ferreira, A., Raposo, A., et al. Creepy tracker toolkit for context-aware interfaces. In: Proceedings of the Interactive Surfaces and Spaces. ISS '17; New York, NY, USA: ACM. ISBN 978-1-4503-4691-7; 2017, p. 191–200. doi:10.1145/3132272.3134113. URL `http://doi.acm.org/10.1145/3132272.3134113`.

[84] Kalantari, N.K., Wang, T.C., Ramamoorthi, R.. Learning-based view synthesis for light field cameras. ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2016) 2016;35(6).

[85] Ribeiro, C., dos Anjos, R.K., Fernandes, C.. Capturing and documenting creative processes in contemporary dance. In: Proceedings of the 4th International Conference on Movement Computing. MOCO '17; New York, NY, USA: ACM. ISBN 978-1-4503-5209-3; 2017, p. 7:1–7:7. doi:10.1145/3077981.3078041. URL `http://doi.acm.org/10.1145/3077981.3078041`.