2
Related Work

In this chapter we present a brief history of IBR and review relevant research results for free viewpoint both in static and dynamic scenes. These works present different strategies for model acquisition, representation and rendering. However, the review is focused especially in model representations and corresponding rendering techniques, since our main goal is visually-accurate rendering at interactive frame rates. We start with real applications that greatly helped spur IBR research.

2.1
IBR real applications

Maybe one of the most popular uses of IBR was the *Bullet Time Effect* in 1999 movie *The Matrix* by Warner Bros [1]. The technique used still cameras surrounding an object in a predefined array, forming a complex curve in space, triggered sequentially or simultaneously. Then, singular frames taken from each of the still cameras were arranged and displayed consecutively to produce an orbiting viewpoint of an action frozen in time or in hyper-slow-motion.

Although the technique used in The Matrix, in theory, allowed for limitless perspectives and variable display frame rates with a virtual camera, those perspectives were limited to the predefined camera paths. Besides, many input cameras and man-hours were necessary to make the virtual camera fly-through smooth and realistic.

But it was more than one decade before *The Matrix*, in the early 1980's, that the *freeze frame* effect was first demonstrated by Tim Macmillan's *Time-slice* [20]. An earlier version consisted of 360 pinhole film cameras arranged in a circle looking towards the center of a circle, where the subject was positioned. Filming was done in the dark, using a flash. A later version reduced the number of cameras to 120, covering 90°.

Another similar approach was used by Dayton Taylor's *Timetrack* system to produce commercials in 1995 [34]: the illusion of moving through a frozen slice of time was produced by rapidly jumping between different still cameras arranged along a path, just like it would be done some years later in *The*

2.1(a): Input cameras arranged in a predefined array.

2.1(b): Frozen-time frame in a fly-through around the character.

Figure 2.1: Bullet-time effect shows the necessity of counterbalancing the number of input cameras and quality of rendered images.

Matrix. Also in 1995, Michel Gondry's "Like a rolling Stone" music video clip innovated by using morphing between adjacent cameras rather than just jumping from one to another.

The *freeze frame/bullet time* effect attracted the interest from the research community. But earliest works in IBR focused unsurprisingly on dealing with static scenes. Pioneering works include Chen and Williams' View Interpolation [8], Chen's QuickTime VR [7], McMillan and Bishop's Plenoptic Modeling [23], Levoy and Hanrahan's Light Field Rendering [19], Gortler et al's Lumigraph [14].

An even more promising application of the method is Free viewpoint TV (FTV) [33]: multi-view video and multi-view depth would be broadcasted, allowing for a free viewpoint experience to the final spectator. Since December 2001 MPEG has been working on the exploration of 3D Audio-Visual (3DAV) technology, and since then has received strong support from TV industry organizations for FTV standardization.

Those works may differ in the number of image samples necessary for obtaining good rendering results, in their representation of the scene, and in the rendering algorithm itself. However, all of them share the general goals of IBR depicted in Figure 2.2: create a representation linked to images of the acquired scene, and composite views to create a new one.

Although early image-based representations that are based solely on image samples, like Light Field Rendering and Panoramas, require very simple rendering techniques, a great number of input samples are necessary. Later on, more sophisticated representations were proposed to deal with the trade-off between images and geometry, and rendering techniques changed accordingly.

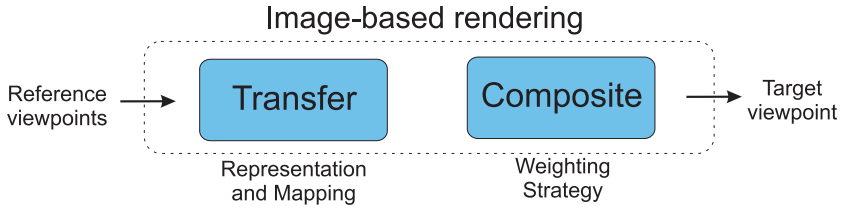

Figure 2.2: IBR goals: establish mapping between representation and image screen, and blend.

2.2 Static scenes

By removing time t and light wavelength λ, in 1995 McMillan and Bishop [23] introduced the concept of Plenoptic Modeling, with the 5D version of the plenoptic function $P5(V_x, V_y, V_z, \theta, \phi)$. An even simpler representation is the 2D panorama, where the viewpoint is fixed ($P2(\theta, \phi)$). It can be cylindrical, as in 1995 Chen's Quicktime VR [7], or spherical, as in 1997 Szeliski and Shum's work [32].

Levoy and Hanrahan's 1996 Light field rendering system [19] constrains the plenoptic function to a bounding box, thus representing it as a 4-dimension function. Rays are interpolated assuming that the scene surface is close to a focal plane. Objects surfaces located far away from the focal plane appear blurred at interpolated views.

Lumigraph system [14], proposed in 1996, uses a similar rendering method, also restricted to a bounding box. However, rather than Light Field's unique focal plane, it uses an approximation of 3D object surface to reduce the blur problem. Still, a huge number of input images are necessary for high-quality rendering.

Chen and Williams' 1993 View Interpolation method [8] makes use of implicit geometry to reconstruct arbitrary viewpoints given two input images and dense optical flow between them. The method works well when input views are close by. Otherwise, the overlapping parts may become too small, impairing the dense optical flow computation.

Also using implicit geometry, Seitz and Dyers's 1996 View Morphing technique [29] reconstructs any viewpoint on the line that links two optical centers of the original cameras. Intermediate views are exactly linear combinations of two views given that the camera motion is perpendicular to the camera viewing direction.

The aforementioned works either require a large number of images for rendering (methods that do not rely on geometry) or require very accurate image registration (methods that use implicit geometry) for high-quality

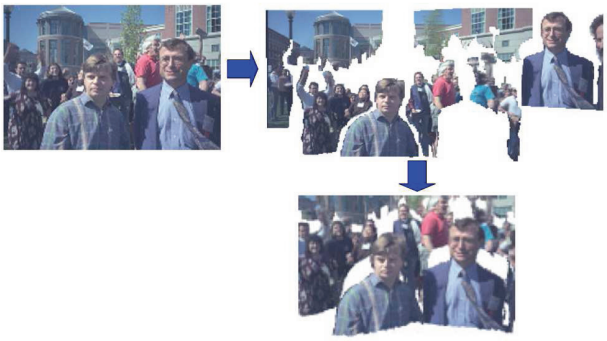

Figure 2.3: Layered Depth Images [30]. Input images (left) used to generate the layered representation of a scene (top right). It allows for reconstruction of views free from disocclusion problems (bottom).

virtual synthesis. Those limitations can be overcome through the usage of explicit 3D information, encoded either in the form of 3D coordinates or depth along lines-of-sight.

In 1999 McMillan [22] argue that 3D warping techniques can be used to render new viewpoints when depth information is available for every point in images. This is accomplished by unprojecting pixels of the original images to their proper 3D locations, and subsequently reprojecting them onto the new viewpoint. The side-effect of that method is the appearance of holes in the warped image.

Difference of sampling resolution (as in the case of zooming-in) or disocclusions, i.e. depth discontinuities, are the causes of holes generation. Splatting [15] has proved to be enough to fill holes introduced by sampling differences, but it cannot deal with disocclusions.

Shade et al's 1998 Layered Depth Images (LDIs) [30], proposed the storage of depth information not only for what is visible in the input image, but also for everything behind the visible surface. In other words, each pixel in the input image contains a list of depth and color values. The correct position in that list could be retrieved and used accordingly depending on the new viewpoint's position. This layered representation can be seen in Figure 2.3.

Another use of explicit geometry in IBR is View-dependent texture-mapping (VDTM), proposed in 1996 by Debevec et al [11], depicted in Figure 2.4. It consists in texture-mapping 3D models of a reconstructed architecture environment, through warping and blending of several input images of that environment. The technique was later improved by Debevec et al [10], in 1998, to reduce computational cost and to allow for smooth blending. The main advantage of that approach is the usage of projective texture mapping, which boosts performance through the usage of graphics hardware.

Regarding the composition process, the Unstructured Lumigraph [5],

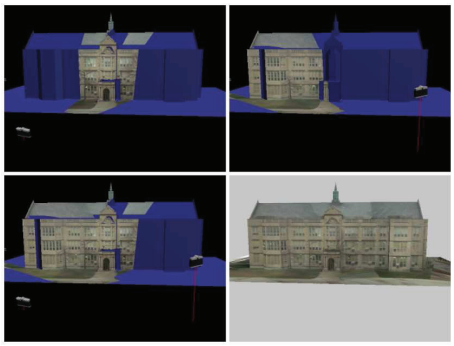

Figure 2.4: View-dependent texture mapping [11]. Input images are projected onto reconstructed architectural model, and assembled to form a composite rendering. Top two pictures show images projected onto model, lower left shows results of blending those two renderings, and lower right shows final result of blending a total of 12 original images.

proposed in 2001 by Buehler et al, presents a very detailed analysis of how textures can be blended based on relative angular position, resolution, and field-of-view. It is a valuable reference for more principled and visually-accurate composition.

Finally, Kang and Szeliski [18] introduced in 2004 the idea of not only using view-dependent textures, but also view-dependent geometries for dealing with non-Lambertian surfaces properties. Warped depth images are blended to produce new views that resemble original non-rigid effects very effectively.

Further research works have focused on how to handle non-rigid effects, but works presented in this section have been successfully adapted to deal with the more intriguing task of rendering dynamic scenes with IBR.

2.3 Dynamic scenes

As mentioned in the previous section, the bullet-time/freeze-frame effect is a very popular application of IBR for dynamic scenes, and its popularity helped spur IBR research on the pursuit of free viewpoint in what is called video-based rendering (VBR) [21].

Extending IBR techniques to dynamic scenes with arbitrary viewpoint selection while the scene is changing is not trivial, although its application is extremely attractive. Associated problems are twofold. First, there are hardware-related issues such as camera synchronization, calibration and images acquisition and storage. Decreasing costs of hardware and technology improvements helped make the capture and subsequent processing of dynamic scenes more practical. Second, it is difficult to achieve automatic generation of seamless interpolation between views for arbitrary scenes. Proposed techniques must deal

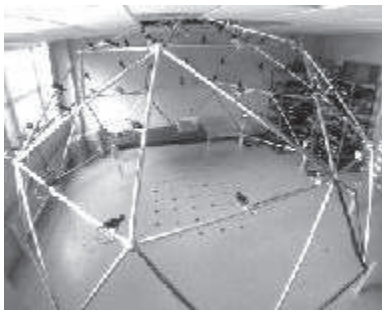

Figure 2.5: Kanade et al's Virtualized Reality geodesic dome [17].

with those difficulties to achieve high-quality rendering at reasonable time.

One of the earliest VBR systems is Kanade et al's 1997 Virtualized Reality [17]. Their architecture involved 51 cameras arranged around a 5-meter geodesic dome, as shown in Figure 2.5. Cameras captured 640x480 video at 30 fps. An important aspect to notice about their work is the two-step video acquisition: real-time recording and an offline digitization step. Virtualized Reality computed a dense stereo depth map for each camera, used as view-dependent geometry for view synthesis. A first version of the system used the closest reference view as a basis, and other two neighbor cameras for hole filling, while a second version involved the merging of depth maps into a single model to be textured with multiple reference views. A version named *Eyevision* was successfully used commercially by CBS Television at Super Bowl XXXV in 2001, with more than 30 cameras involved.

Vedula et al [36] extended Virtualized Reality in 2005 by employing spatio-temporal view interpolation. It explicitly recovered 3D scene shape at every time frame and also 3D scene flow (local instantaneous 3D non-rigid temporal deformation). A voxelization algorithm was used for both 3D shape extraction and rendering. For novel view generation, ray-casting along with blending weights were used. Weights were a combination of temporal and spatial proximity to the novel viewpoint.

Stanford Light Field Camera was proposed initially in 2002 with 6 input cameras [37]. It was later extended in 2004 to a system with 128 CMOS cameras [35], designed based on the IEEE 1394 high speed serial bus (Firewire). Cameras are capable of acquiring 640x480 videos at 30 fps, with 8:1 MPEG compression.

Goldlücke et al [13] in 2002 used a subset of Stanford Light Field Camera for acquiring and displaying dynamic scenes. In their work, cameras calibration is done for extrinsic and intrinsic parameters estimation, to reduce radial distortion and also to reduce color and brightness variation across cameras.

Depth maps are obtained through depth from stereo. After depth es-

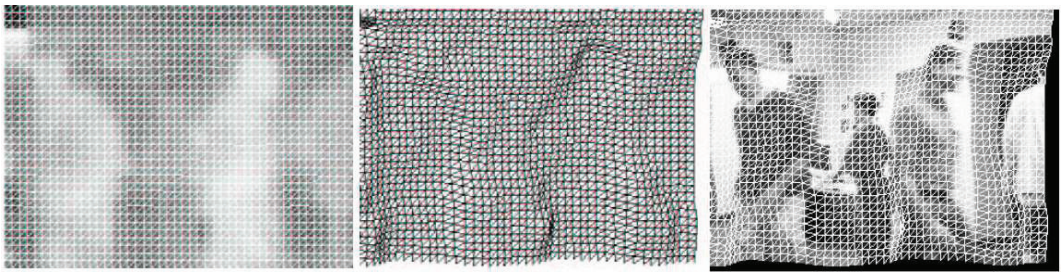

Figure 2.6: Goldlücke results [13]. The regular triangular mesh causes inaccurate appearance at the vicinity of depth discontinuities.

timation for all images and timeframes, interactive rendering is achieved by employing 3D warping. A regular, downsampled triangle mesh is created covering each of the input depth images. A vertex program is used for warping to the novel view, and composition of 4 different reference views is done through weights based on proximity to the novel view: the closer the input image, the higher its weight.

They report a frame rate of 11 fps with a mesh resolution of 160x120. Figure 2.6 shows how the triangular mesh superimposed on a reference view's depth map. Since the triangular mesh is continuous and regular, at the vicinity of big depth discontinuities the appearance is usually incorrect. In fact, the mesh downsampling generates an unpleasant blurring effect at objects boundaries. A great improvement regarding rendering quality is presented by Zitnick et al [38]. Although their system is quite modest in size, with only 8 cameras, higher resolution images (1024x768) are captured at 15 fps. Photorealism is achieved using a two-layer representation inspired by Layered-depth images [30], mentioned in the previous section.

Their system calculates a dense depth map for each input color image with their proposed algorithm. After that, they divide the scene representation in two layers: boundary layer B, around depth discontinuities, and main layer M. To generate this representation, a variant of Bayesian matting [9] is used to automatically estimate foreground and background colors, depths and opacities around depth discontinuities.

System configuration is shown in Figure 2.7. At rendering time, the two reference views nearest to the novel view are chosen, warped through usage of a custom vertex shader into separate buffers, and finally blended through a custom fragment shader that calculates contribution weights based on angular proximity of the reference view to the novel view. Their system involves both offline and real-time phases. Computation of depth maps, boundaries identification and matting in those areas, compression and storage are offline processes. Decoding and rendering are done in real-time, with reported per-

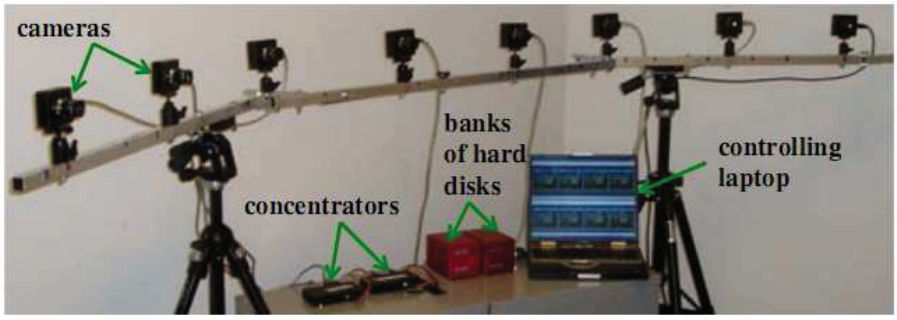

Figure 2.7: Camera setup in Zitnick et al [38]. Eight cameras are used to capture 1024x768 images, synchronized with commissioned PtGrey concentrator units.

Figure 2.8: Rendering results for Zitnick et al [38]: (a) main layer M from one view rendered, with depth discontinuities erased; (b) boundary layer B rendered; (c) main layer M for other view rendered; (d) final blended result.

formance of 5 fps for 1024x768 images. It yields the best results among all mentioned VBR systems, with examples of generated views depicted in Figure 2.8.

Our rendering method also relies on 3D warping and blending of a pair of reference views, but assumes that depth maps are previously calculated: we focus on the rendering stage, not dealing with depth map estimation.

The objective of this work is to completely avoid offline processes like matting, but still yield high-quality rendering of virtual views. We intend to use solely depth images (color image + depth map) as input for our algorithm. Our contribution is a set of techniques for warping and blending views which run entirely on the GPU.