A Survey of Image-based Rendering Techniques

Sing Bing Kang

Cambridge Research Laboratory
Technical Report Series

CRL 97/4

August 1997

Cambridge Research Laboratory

The Cambridge Research Laboratory was founded in 1987 to advance the state of the art in both core computing and human-computer interaction, and to use the knowledge so gained to support the Company's corporate objectives. We believe this is best accomplished through interconnected pursuits in technology creation, advanced systems engineering, and business development. We are actively investigating scalable computing; mobile computing; vision-based human and scene sensing; speech interaction; computer-animated synthetic persona; intelligent information appliances; and the capture, coding, storage, indexing, retrieval, decoding, and rendering of multimedia data. We recognize and embrace a technology creation model which is characterized by three major phases:

Freedom: The life blood of the Laboratory comes from the observations and imaginations of our research staff. It is here that challenging research problems are uncovered (through discussions with customers, through interactions with others in the Corporation, through other professional interactions, through reading, and the like) or that new ideas are born. For any such problem or idea, this phase culminates in the nucleation of a project team around a well articulated central research question and the outlining of a research plan.

Focus: Once a team is formed, we aggressively pursue the creation of new technology based on the plan. This may involve direct collaboration with other technical professionals inside and outside the Corporation. This phase culminates in the demonstrable creation of new technology which may take any of a number of forms - a journal article, a technical talk, a working prototype, a patent application, or some combination of these. The research team is typically augmented with other resident professionals—engineering and business development—who work as integral members of the core team to prepare preliminary plans for how best to leverage this new knowledge, either through internal transfer of technology or through other means.

Follow-through: We actively pursue taking the best technologies to the marketplace. For those opportunities which are not immediately transferred internally and where the team has identified a significant opportunity, the business development and engineering staff will lead early-stage commercial development, often in conjunction with members of the research staff. While the value to the Corporation of taking these new ideas to the market is clear, it also has a significant positive impact on our future research work by providing the means to understand intimately the problems and opportunities in the market and to more fully exercise our ideas and concepts in real-world settings.

Throughout this process, communicating our understanding is a critical part of what we do, and participating in the larger technical community—through the publication of refereed journal articles and the presentation of our ideas at conferences–is essential. Our technical report series supports and facilitates broad and early dissemination of our work. We welcome your feedback on its effectiveness.

Robert A. Iannucci, Ph.D.
Director

A Survey of Image-based Rendering Techniques

Sing Bing Kang

August 1997

Abstract

In this technical report, we survey the techniques for image-based rendering. In contrast with the conventional approach of using 3-D models for creating new virtual views, image-based rendering techniques require only the original images to produce new virtual views. We have identified four basic categories for image-based rendering techniques, namely non-physically based image mapping, mosaicking, interpolation from dense samples, and geometrically-valid pixel reprojection. This division is based primarily on the nature of the pixel indexing scheme. We discuss the characteristics of each of these categories and present representative techniques. From this survey, it is interesting to note that image-based rendering has evolved to be an area that has generated active interest from both computer vision and computer graphics communities.

CRL Technical reports are available on the CRL's web page at
http://www.crl.research.digital.com.

Digital Equipment Corporation
Cambridge Research Laboratory
One Kendall Square, Building 700
Cambridge, Massachusetts 02139

Contents

List of Figures

List of Tables

1 Introduction

The traditional approach for generating virtual views of an object or a scene is to render directly an appropriately constructed 3-D model. The 3-D model can be produced using a CAD modeler or from real data (such as those obtained using 3-D digitizing tools, active rangefinders, or stereo). The forms that the 3-D models can assume are polygonal, bicubic parametric patches, constructive solid geometry, and space subdivision representations (such as the octree) [49]. Realism can be enhanced by applying texture maps or environment maps (images of real or synthetic scenes), shading algorithms (e.g., Gouraud [13] or Phong [38]), bump maps [4], or a combination of these techniques on surfaces of 3-D models.

Volume rendering or visualization of voxel-based data (as opposed to surface-based data mostly referred to earlier) is also another type of 3-D model rendering. The voxel-based data may be actual MRI data slices for biomedical applications or mathematical data resulting from material stress simulation for engineering applications. Depending on the application, the volume may be rendered as an explicit surface representation recovered using, say, the Marching Cubes algorithm [29]. It may also be rendered with the voxels within the volume data being assigned different opacities and/or color according to their classification or properties.

More recently, there is an emerging and competing means of creating virtual views called *image-based rendering*. In contrast with 3-D model-based rendering, image-based rendering techniques rely primarily on the original or trained set of images to produce new, virtual views.

Comparisons between the 3-D model-based rendering and the image-based rendering techniques are shown in Table 1. In 3-D model-based rendering, 3-D objects and scenes are represented by explicitly constructed 3-D models (from CAD modeler, 3-D digitizer, active range, or stereo techniques). New virtual views are produced by repositioning these models or the virtual camera and rendering. On the other hand, in image-based rendering, objects and scenes are repre-

3-D model-based rendering	Image-based rendering
Explicit use of 3-D models	Directly uses collection of images
Uses conventional rendering pipeline	Based on interpolation or pixel reprojection
Speed dependent on scene complexity	Speed independent of scene complexity
Relies on hardware accelerator for speed	Relies on processor speed
Requires sophisticated software for realism	Realism depends on input images

Table 1: Comparison between 3-D model-based rendering and the image-based rendering techniques

sented by a collection of images, which is generally easier to acquire for real objects and scenes.

3-D model-based rendering relies on the conventional rendering pipeline that includes modeling transformation, view transformation, culling, and hidden surface removal. As such, the cost of rendering is dependent on the object or scene complexity, specifically the number of facets or voxels used in the representation. For fast rendering of more complicated scenes, a 3-D graphics accelerator would be required. In addition, expensive software is usually required to produce photorealistic images.

In contrast, image-based rendering techniques rely on interpolation using the original set of input images or pixel reprojection from source images onto the target image in order to produce a novel virtual view. The cost of rendering is independent of the scene complexity. As a result, while these techniques do not require specialized graphics accelerators, they may incur moderate to high amount of computation. The use of potentially many input images (especially to represent many wide scenes) would result in significant memory requirement. On the other hand, the amount of realism in the synthesized image depends on the quality of the input images. Photorealistic synthesis of images is possible from photographs of real scenes.

There are four distinct (but not necessarily mutually exclusive) categories of image-based rendering techniques. These categories are created based primarily on the nature of the scheme for pixel indexing or transfer. They are: *non-physically based image mapping*, *mosaicking*, *interpolation from dense samples*, and *geometrically-valid pixel reprojection*.

2 Non-physically based image mapping

The first category of non-physically based image mapping is generally rather straightforward. In this category of techniques, 3-D geometry is not considered at all in the pixel location computation. Two types of non-physically based image mapping can be identified, namely direct and indirect (learned) image mapping.

Images created using direct image mapping techniques originate from pairs of possibly unrelated images. These techniques are used most widely in the advertising and entertainment industries. Possibly the most well-known technique is that of Beier and Neely's feature-based morphing [3], where the initial manual image correspondence is done using pairs of oriented lines. This method was used to produce the image morphing sequence in Michael Jackson's "Black or White" music video. The morphing process involves warping two images so that the source shape slowly assumes the target shape, with the attendant image mixing, or cross-dissolving between the two. Warping is done through the process of inverse mapping, i.e., scanning each pixel in the target

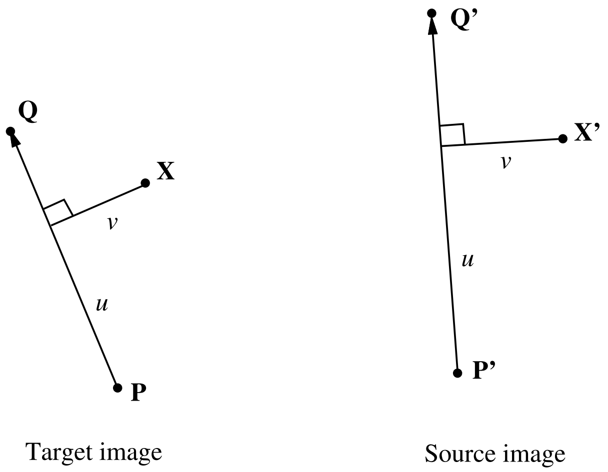


Figure 1: Definitions of u and v for a line pair (adapted from [3]).

image and finding the pixel in the source image that corresponds to it. Pixel shifts are calculated such that pixels that fall exactly on each manually designated line in the target image are mapped to pixels on the corresponding line in the source image. Every other pixel position is computed using a weight that is a function of the:

- projected distance u (relative to the length of the line) along each line,
- perpendicular distance v from each line, and
- parameters that indicate the strength of influence of that line.

u and v are defined in Figure 1.

While this feature-based interpolation technique allows the animator more flexibility in explicitly defining feature correspondences, it suffers from lack of control over areas outside of the designated features and the high computational requirement.

Several other image warping techniques and examples of morphing are given in Wolberg's book [50]. An example of a warping technique is the 2-D spline mesh mapping. In this technique, the user specifies separate grids of points in both the source and target images. Each target grid point has a corresponding source grid point. Displacements between grid points can be computed using some form of interpolation; linear interpolation is the easiest and hence most popular. For implementational efficiency, image deformation can be achieved using a 2-pass algorithm, where the first pass involves a transform in the x-direction, followed by a transform in the y- direction.

In comparison to the feature-based interpolation technique, the 2-D spline mesh technique allows local control and faster speeds. However, it is more difficult using the mesh technique to produce the desired warping. In addition, if the mesh technique is a 2-pass algorithm, there is a possibility of failure for significant rotational distortions (say, 90° in any direction).

The more recent set of non-physically based image mapping techniques, namely those of indirect or learned image mapping, uses a training set of specific kinds of images to produce novel views. Examples of specific sets of images are those of human faces [47] or human stick figures [20]. The main idea is to represent an arbitrary view of a 3-D object as a linear combination of a set of original disparate views, considered to be *basis* views. In [47], an image of a 3-D face at an arbitrary camera viewpoint is considered to be a linear combination of images of different faces taken at the same facial pose (see Figure 2). Hence, in order to generate novel views of a new face not in a training set, we would first have to extract the best set of linear weights associated with the face images at a standard pose in the training set. These weights are computed such that the weighted average of the face images at a standard pose in the training set is the best approximation to the new reference face image at the same standard pose (according to some metric, usually based on error in intensity). The linear weights can be extracted if correspondence between the new face and the faces in the training set has been done. The principle of this method is illustrated in Figure 2. The "learned" set of linear weights ($a_1 ... a_N$ in Figure 2) can then be used to generate a novel view of the new face at a desired pose. This is all done without explicitly knowing its 3-D structure. Using a similar principle, novel stick figures can be generated from a set of prototypes (Figure 3).

3 Mosaicking

The term "mosaics" refer to the combination of at least two different images to yield a higher resolution or larger image. The mosaic, which has a wider field of view than its constituent images, is a more compact representation that allows views within it to be quickly generated and viewed. Early work at mosaics are mostly those of aerial or satellite pictures of Earth or in astronomy. While many image mosaics are produced without removing visible mosaic boundaries, there exist a number of techniques to reduce or remove these discontinuities. They range from computing the linear ramp function to obtain equal values at mosaic boundaries [32], histogramming the overlap areas to find the gray-level shift [33], applying an iterative method for recovering a spatially smooth "seam-eliminating function" that minimizes errors in the overlap areas [36], to the multiresolution spline technique [6]. In the multiresolution spline technique, at each resolution level, band-pass

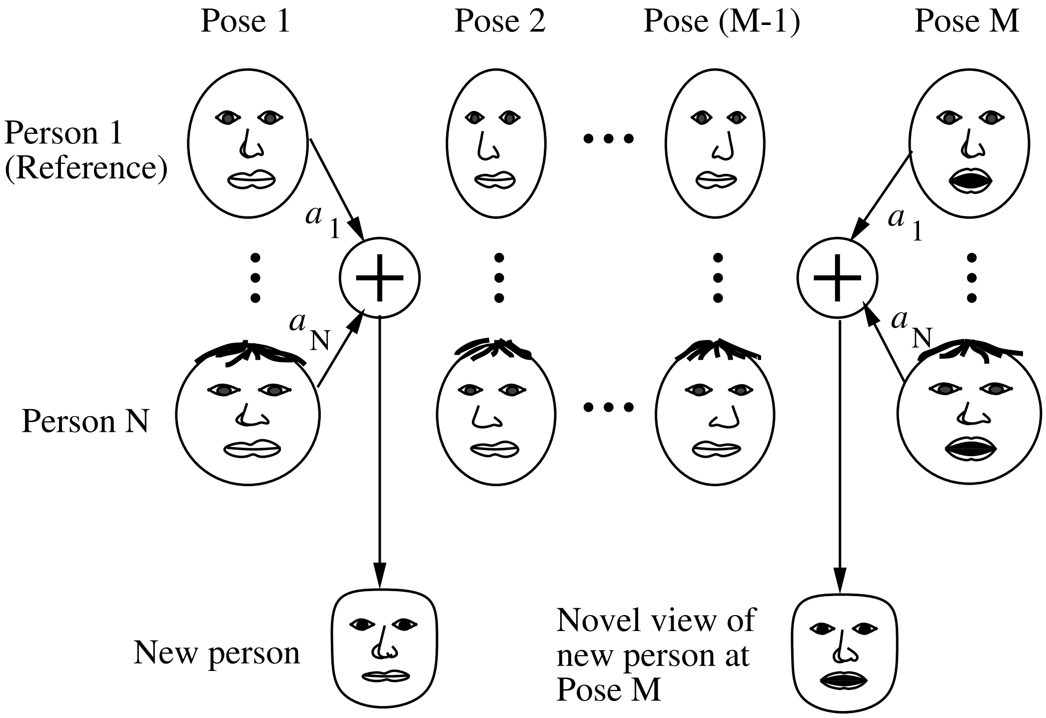
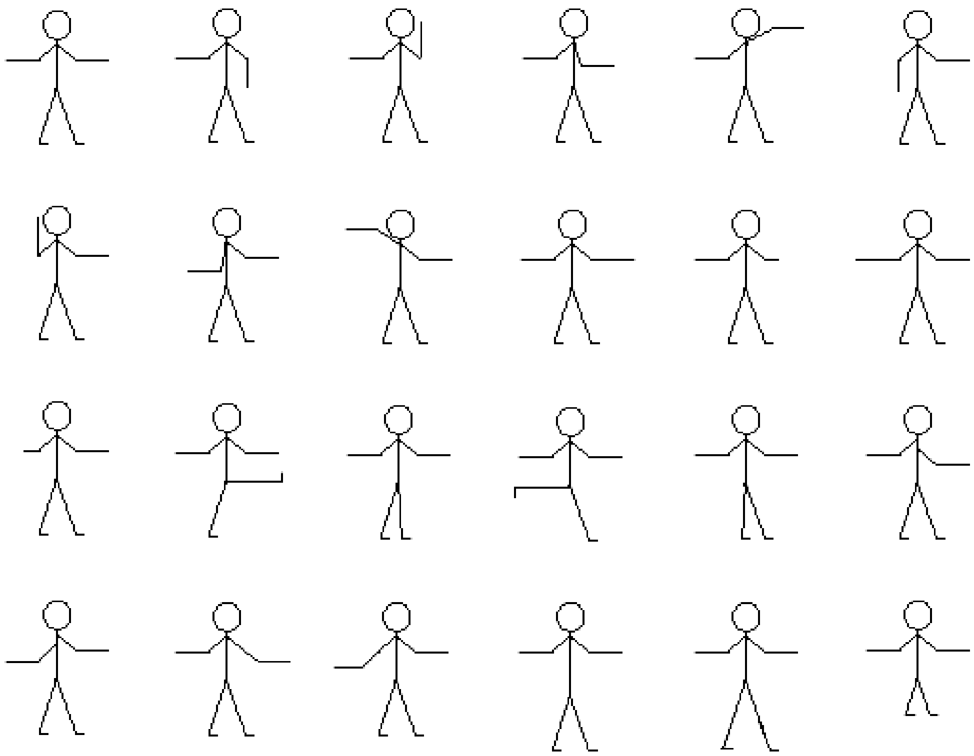


Figure 2: Illustration of the linear object class concept in [47]. The training set is composed of faces of different people (along the vertical axis) assuming the same different poses (horizontally across). One person in the training set is taken to be the reference for all the poses. The weights $a_1 ... a_N$ are computed based on the "standard" pose only (left). These weights are then used to generate a novel view of the new face (right).



Set of prototype human stick figures

Novel stick figures generated from the prototypes

Figure 3: Stick figure example (taken from [19]).

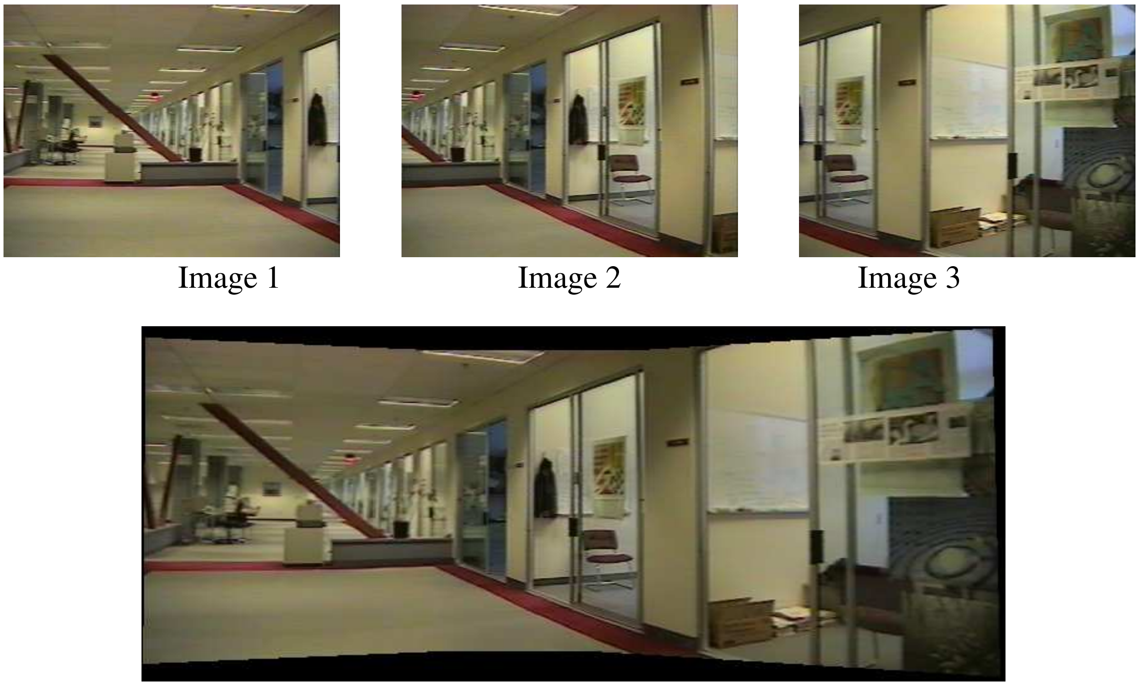


Image 1 Image 2 Image 3

Figure 4: Open space scene: (Top) Three original images; (Bottom) Panorama of open space after compositing.

filtered images are composed by way of weighted averaging with a transitioning zone whose width is dependent on the current resolution [6]. The final mosaic is computed by combining the mosaics at all resolution levels.

Composing a series of aerial pictures is easy because of the minimal perspective distortion. Systematic techniques have subsequently been developed to compositing images of relatively more nearby scenes with noticeable perspective distortion. They are used to create larger images, i.e., rectilinear panoramic images (e.g., [16, 44, 42]), super-resolution images (e.g., [17]), image mosaics with identified multiple motions (e.g., [39, 48]), and image mosaics with identified parallax (e.g., [25]). An example of a sequence of images (taken while rotating the camera about its projection center) is shown at the top of Figure 4; their corresponding mosaic is shown at the bottom of Figure 4.

Particularly relevant to the notion of image mosaics with multiple motions is Microsoft's Talisman system [46]. In a departure from conventional 3-D graphics rendering architectures, Microsoft's Talisman [46] is a 3-D graphics and multimedia hardware architectural design that treats individually animated objects as independent image layers. Prior to compositing these layers, a full

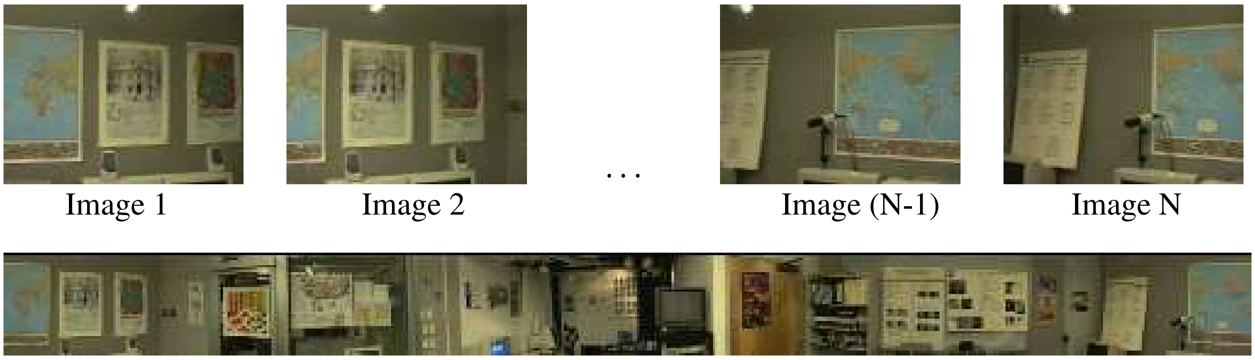


Figure 5: Lab scene: (Top) Undistorted and cylindrically projected image sequence; (Bottom) Panorama of lab scene after compositing.

2-D affine transformation is applied to each layer to simulate 3-D rendering and exploit temporal image coherence.

While rectilinear panoramic images are convenient representations for a relatively small field of view (less than 180°), they are mathematically problematic for very wide scenes. In such circumstances, either cylindrical or spherical representations are more suitable. However, for these representations, the section of interest within the image representation has to be undistorted prior to viewing. In creating panoramic images, the camera focal length has to be determined first, usually through calibration. The original image sequence will then be warped from rectilinear to cylindrical coordinates before being mosaicked. An example of such a sequence is shown at the top of Figure 5, and the resulting cylindrical panoramic image is shown at the bottom of Figure 5.

Cylindrical panoramic images have been created for a variety of purposes, from fixed-location visualization [9] to variable location visualization ("plenoptic modeling") [31] to recovery of 3-D model (omnidirectional multibaseline stereo) [21]. There are also a number of commercial cylindrical panoramic viewers, which include Infinite Pictures' SmoothMove, IBM's PanoramIX, Visdyn's Jutvision, Apple's QuickTime VR$^{\mathrm{TM}}$ [9], and RealSpace, Inc.'s RealVR$^{\mathrm{TM}}$.

The drawback of using a cylindrical panoramic image is its inability to include parts of the scene at the top and bottom. This deficiency can be overcome by using a spherical image. A spherical image can be created by merging multiple images taken using a camera with a fisheye lens [51], or by merging many images taken with a camera with a normal lens at different tilt and pan orientations [45]. In either case, care must be taken to ensure that the projection centers associated with all the constituent views coincide if the resulting mosaic is to be physically exact. A commercial spherical image creation and viewing software is available in the form of Interactive Pictures Corp.'s IPIX (formerly Omniview's Photobubble). An IPIX image is created from two photographs captured by a fisheye lens at opposing viewpoints and subsequently stitched together. As in the case of the cylindrical panoramic image, the portion of the spherical image to be viewed has to be undistorted (i.e., undergo perspective correction) first.

The panoramic image created is theoretically exact only under two conditions: (1) the scene is of arbitrary shape while the camera motion is a rotation about its center of projection, or (2) the scene is planar with the camera motion being arbitrary. (This is assuming that the nonlinearities of the camera optical system such as the radial distortion parameters are known and accounted for. In the case of the cylindrical panoramic image, the camera intrinsic parameters such as the camera focal length are assumed known.) If any of these conditions is not satisfied, then the resulting mosaic will not be physically correct. However, for some applications, this is acceptable. A fast method for generating a mosaic through *manifold projection* [37] has been proposed. In this method, a mosaic is created by merging central strips of each image as the camera moves along an arbitrary trajectory. A similar technique has been adopted by Zheng and Tsuji to create panoramic representations for robot navigation [54]. Szeliski and Shum [45] use all parts of the images in a sequence to produce a panoramic image. For a sequence of images whose camera centers may not coincide, they apply a *deghosting* technique based on image subdivision and patch alignment to minimize the effects of misregistration due to parallax. As before, the resulting mosaic will not be physically exact.

In order to create the image mosaics from multiple images, the constituent images have to registered first. The image registration techniques that have been used include:

1. Manual
   The operator either manually positions each image relative to the others or manually picks corresponding points that enable the program to compute the global image displacements.

2. Exhausive search
   The displacement may be determined by searching over a limited range of motion.

3. Known camera positions and optional local registration
   The position of the camera will be known if it is digitally controlled; the local registration performs iterative local gradient descent to minimize the difference between the overlapped areas. The local registration step may employ a hierarchical scheme to account for larger errors in motion. The local registration step may be necessary if the camera positions are known only approximately.

4. Global (phase correlation [24]) and local (iterative)
   If the position of the camera is unknown and there is sufficient texture in both images, a

global registration technique in the form of phase correlation may be used. It computes the 2-D Fourier transform of both images and finds the difference in their phases, which maps to 2-D displacement.

5. Hypothesize from image features
The idea is to detect specific image features, hypothesize the correspondences between the features, and compute the associated 2-D projective transform. This would give rise to an error in difference in overlap areas. The correct displacement is the one corresponding to the minimum error. An example is to use geometric corners [55], which has been shown to work even with large rotations and significant displacements.

The alternative to creating the mosaics in software is to customize hardware to produce the same effect. Some of the customized camera systems are

- Digitally controlled slowly rotating camera with panoramic images created by combining vertical slits from each camera position [18].
- Camera system with conic mirror setup (COPIS - COnic Projection Image Sensor) [52].
- Interactive Pictures Corp.'s IPIX (formerly Omniview's Photobubble) that uses a fish-eye lens. A spherical image is created by stitching together two hemispherical images.
- Omnidirectional camera with hyperboloid mirror [53].
- Omnidirectional camera with paraboloid mirror [35].
- Panoramic camera system [34] which uses four CCD cameras aimed upward at four triangular mirrors. Special software reverses the mirror images and blends the individual pictures seamlessly into a single image on the monitor. The system displays seven and a half panoramic frames per second.

4 Interpolation from dense samples

As its category name implies, the idea of this class of methods is to first build up some form of lookup table by taking many image samples of an object or a scene from many different camera viewpoints. Subsequently, the image associated with any given arbitrary viewpoint is synthesized by interpolating the stored lookup table. The advantage of this class of methods is that unlike all other methods, pixel correspondence, which can be difficult, is not necessary. In addition, they encapsulate within the lookup table the appearance of the object or scene at actual discrete camera viewpoints; in other words, the lookup table is an approximation of the *plenoptic function* [1]. (The plenoptic function is a 5-D description of the flow of light at every 3-D position and 2-D viewing direction. The flow of light is independent of the roll about the viewing direction.) Because the process of image synthesis is based on the interpolation of the light field lookup table, fast speeds in visualization have been achieved. On the negative side, these methods require intensive data acquisition, high storage and memory requirements, as well as the knowledge of the camera viewpoints at every sample during data acquisition. The last requirement of knowing the camera viewpoints may not be easily satisfied in practice for large scenes.

Note that in comparison to the indirect or learned image mapping technique (part of non-physically based image mapping category), the number of samples is clearly significantly larger (hundreds or even thousands, as opposed to only tens per subject, if not less). In addition, in the current category, the virtual object or scene viewpoint is computed based on samples of the *same object or scene*. This is not necessarily so for the learned interpolation technique. Finally, the learned interpolation technique requires some form of image correspondence between images during the training stage and possibly in the estimation of the virtual view. This is not necessary for techniques in the category of interpolation from dense samples.

Two approaches have been described recently, that of *light field rendering* [27], and the *lumigraph* [12]. Both approaches use a 4-D parameterization of viewing position and direction, namely (u, v, s, t). This representation is called the *light slab* by Levoy and Hanrahan and *lumigraph* by Gortler *et al*. This representation is depicted in Figure 6.

The interpolation scheme used by Levoy and Hanrahan approximates the resampling process by simply interpolating the 4-D function from the nearest samples. They have investigated the effect of using nearest neighbor, bilinear interpolation on the $u-v$ plane, and full 4-D quadrilinear interpolation.

In Gortler *et al*.'s approach, a continuous lumigraph is reconstructed as a linear sum of the product between a basis function and the value at each grid point in (u, v, s, t). The virtual viewpoint can then be estimated from this continuous lumigraph, either through pixel-by-pixel raytracing or through patch-based texture-mapping operations. Geometric information can be used to guide the choice of the basis functions. In addition, the lumigraph can tolerate a lower sampling density if approximate geometric information is available.

A more restricted version of light field rendering and the lumigraph has been developed by Katayama *et al.* [22]. The motion of the camera during data acquisition is constrained to move along a straight path at fixed increments. Their view synthesis by interpolation is based on the

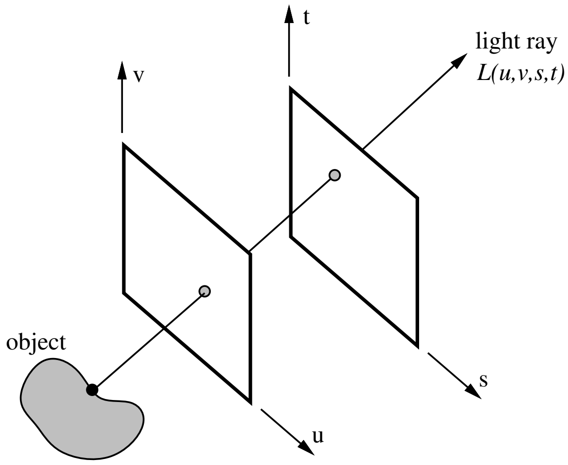


Figure 6: The representation of light field. The center of the (u, v) plane may coincide with the centroid of the object to be visualized.

idea of the epipolar volume [5], where images are stacked on top of each other to create a 3-D slab in (u, v, t), with t being the temporal component. However, as a result of the constrained camera motion, distortions would be evident under virtual viewpoints away from the line of motion.

As indicated earlier, the primary disadvantage of this class of methods is the requirement of a large set of image samples to produce the lookup table. The range of field of view using this approach is currently relatively small. It is not clear if these methods are feasible to view a wide *real* scene, due to the data acquisition requirements (large sample density and assumed known camera viewpoints for all samples).

5 Geometrically-valid pixel reprojection

This class of image-based rendering techniques is also known in the photogrammetric community as *transfer* methods. They are characterized by the use of a relatively small number of images with the application of geometric constraints (either recovered at some stage or known *a priori*) to reproject image pixels appropriately at a given virtual camera viewpoint. The geometric constraints can be of the form of known depth values at each pixel, *epipolar constraints* between pairs of images, or *trilinear tensors* that link correspondences between triplets of images.

If the depth value at each pixel is known, then the change in location of that pixel is constrained in a predictable way. In addition, ordering information is directly known, thus solving the object occlusion and disocclusion problem. Chen and Williams [8] use range data (assumed known *a priori*) to establish correspondences between images. They subsequently perform quadtree decomposition of the image based on offset vectors, or the morph map. Offset vectors are interpolated linearly to create intermediate views. (These intermediate views will be physically exact only if the camera motion is perpendicular to the camera viewing axis). Visibility is computed based on the depth ordering of pixels. This is somewhat similar to Chang's work [7], which instead uses image sequences of real scenes obtained under approximately known camera trajectories. Depth values at each pixel are estimated using a stereo algorithm.

As mentioned before, intermediate views are exactly linear combinations of two views only if the camera motions associated with these intermediate views are perpendicular to the camera viewing direction. This is the driving principle of Seitz and Dyer's work [40]. Given any pair of images whose correspondences are manually extracted, they perform a series of operations to compute an intermediate view, namely,

1. Prewarp the images so that corresponding scan lines are parallel. This operation is also called image rectification. After rectification, the new images constitute different parts (that may overlap) of the same plane.

2. Linearly interpolate the image to yield a rectified intermediate image.

3. Postwarp or "unrectify" this intermediate image.

It is obvious that this method is restricted to computing intermediate views along the line segment that links the two camera projection or optical centers.

5.1 The fundamental matrix

In order to prewarp or rectify the pairs of images (step 1 above), a 3×3 matrix called the *fundamental matrix* [11] has to be computed. If a point $\mathbf{p}_i = (u, v, 1)^{\mathrm{T}}$ in image 1 and a point $\mathbf{p}'_i = (u', v', 1)^{\mathrm{T}}$ in image 2 correspond to the same physical 3-D point in space, then for that pair of images,

$$(u', v', 1)\mathbf{F} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = 0 \tag{1}$$

F is called the fundamental matrix and is of rank 2, and $F = [\mathbf{t}]_\times A$, where A is an arbitrary 3×3 matrix. $\mathbf{t}$ is the change in the camera position between the two images, and $[\mathbf{t}]_\times$ is the matrix form of the cross product with $\mathbf{t}$.

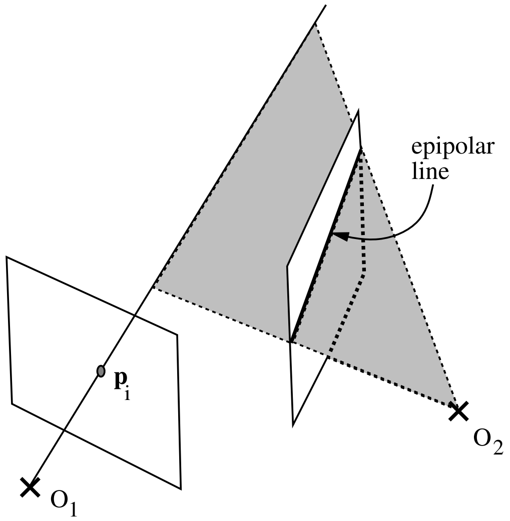


Figure 7: The epipolar constraint on a pair of images. The $\times$'s are the camera optical centers. For a given point in one image, the location of the corresponding point in the other image is constrained to lie on a line called the epipolar line.

The matrix F defines the epipolar geometry between the two images, which constrains the correspondence between points across the two images. This constraint is shown in Figure 7. More details on the camera geometry and its related terminology can be found in [10]. F can be determined from just a collection of image correspondences, without knowing the camera intrinsic parameters, specifically the focal length, aspect ratio, image axis skew, and principal point. In other words, the camera is *weakly calibrated*. If these parameters are known, then we have a more specialized form of F called the *essential matrix* [28]. We can decompose the essential matrix directly into the rotational and translational component of the camera motion. In this case, the camera is *strongly calibrated*.

The eight-point algorithm [28] can be used to compute either the fundamental matrix or the essential matrix, though other methods also exist. A more stable version of the eight-point algorithm that involves pixel coordinate normalization [14] can be used to compute the fundamental matrix.

Laveau and Faugeras [26] use a collection of images called reference views and the principle of the fundamental matrix to produce virtual views. The new viewpoint, which is chosen by interactively choosing the positions of four control image points, is computed using a reverse mapping or raytracing process. For every pixel in the new target image, a search is performed to locate the pair of image correspondences in two reference views. The search is facilitated by using the epipolar constraints and the computed dense correspondences (also known as image disparities)

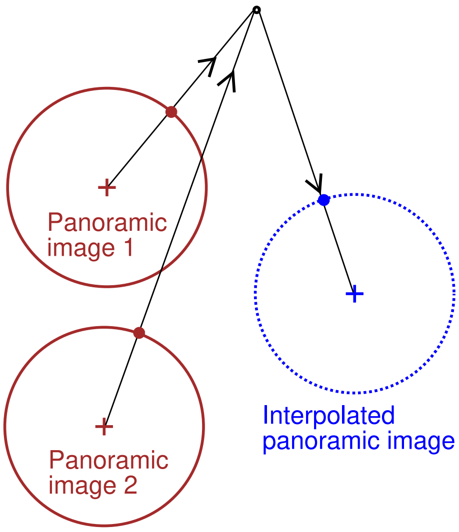


Figure 8: Cylindrical panoramic interpolation.

between the two reference views.

Note that if the camera is only weakly calibrated, the recovered viewpoint will be that of a projective structure. This is because there is a class of 3-D projections and structures that will result in exactly the same reference images. Since angles and areas are not preserved, the resulting viewpoint may appear warped. Knowing the internal parameters of the camera removes this problem.

5.2 Interpolation of cylindrical panoramic images

The idea used in computing new views from rectilinear reference images can be extended to cylindrical panoramic images. There are, however, two important differences: (1) the epipolar lines for cylindrical panoramic images are sinusoidal rather than straight lines, and (2) the camera intrinsic parameters are known (otherwise it would not be possible to create the cylindrical panoramic images). Factor (1) results in higher computation while factor (2) results in simplier recovery of relative camera pose, and guarantees correct virtual shape appearance as long as image correspondences are correct. The basic concept for interpolating cylindrical panoramic images is shown in Figure 8. It is equivalent to computing the 3-D points from image correspondences and projecting them to a new target image. McMillan and Bishop [31] devised an efficient means of transferring known image disparity values between cylindrical panoramic image to a new virtual view. Their approach uses the intermediate quantity called the *generalized angular disparity*, which is analogous to the classical stereo disparity for rectilinear images. They also use an algorithm that guarantees back-to-front ordering that handles the visibility problem.

An image-based rendering tool that we have developed that uses as input cylindrical panoramic images is shown in Figure 9. Dense image correspondences between the panoramic images are established using the spline-based registration technique [43]. The relative camera positions are computed using the eight-point algorithm. The recovered camera positions allow us to then recover the 3-D structure [21], which is displayed at the bottom left of Figure 9. Note that this structure is used for visualization only; it is not used in the computation of the new view. The new view is computed using an idea similar to McMillan and Bishop's.

5.3 The trilinear tensor

The fundamental matrix defines the epipolar constraints between two arbitrary images. Three arbitrary images satisfy multi-linear matching constraints, or trilinearities, as described in [41]. These constraints can be represented by a $3 \times 3 \times 3$ *trilinear tensor* α_i^{jk}, with $i, j, k = 1, 2, 3$. Point correspondences across three images ($\mathbf{p}$, $\mathbf{p}'$, and $\mathbf{p}''$) are linked by the trilinear tensor in the following manner (using the Einstein summation convention):

$$p^i s_j^\mu r_k^\rho \alpha_i^{jk} = 0 \tag{2}$$

with $\mu, \rho = 1, 2$, s_j^1 and s_j^2 representing two lines intersecting at $\mathbf{p}'$, and r_k^1 and r_k^2 representing two lines intersecting at $\mathbf{p}''$. There are four separate constraints associated with (2), each constraint corresponding to a different combination of s_j^1 or s_j^2 and r_k^1 or r_k^2. The geometric interpretation of (2) for the 3-D ray passing through $\mathbf{p}$ and the planes passing through s_j^1 and r_k^1 is shown in Figure 10.

If a trilinear tensor is known for a set of three images, then given a pair of point correspondences in two of these images, a third corresponding point can be directly computed in the third image without resorting to any projection computation. This idea has been used to generate novel views from either two or three reference images [2].

The idea of generating novel views from two or three reference images is rather straightforward. First, the "reference" trilinear tensor is computed from the point correspondences between the reference images. In the case of only two reference images, one of the images is replicated and regarded as the "third" image. If the camera intrinsic parameters are known, then a new trilinear tensor can be computed from the known pose change with respect to the third camera location. The new view can subsequently be generated using the point correspondences from the first two images and the new trilinear tensor. This technique is diagrammatically depicted in Figure 11.

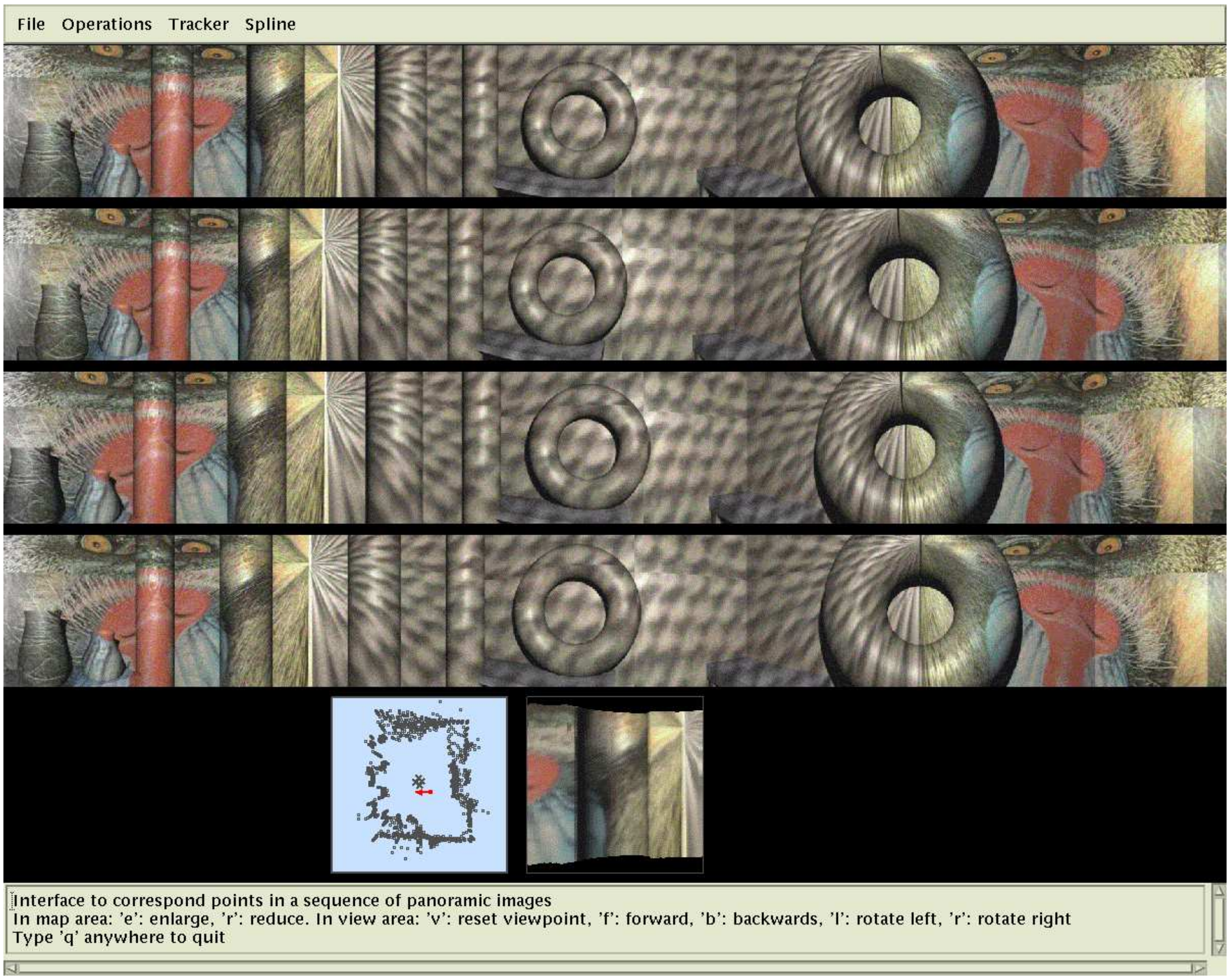


Figure 9: An image-based rendering visualization tool. The top four panoramic images are the input images of a synthetic 3-D scene. The bottom left figure is the top view of the 3-D structure of the scene with the crosses indicating the computed positions of the camera. The point with the arrow indicates the current location of the virtual camera with its direction. The bottom right figure shows the view seen by the virtual camera. Note that no explicit 3-D model is used to generate the virtual image.

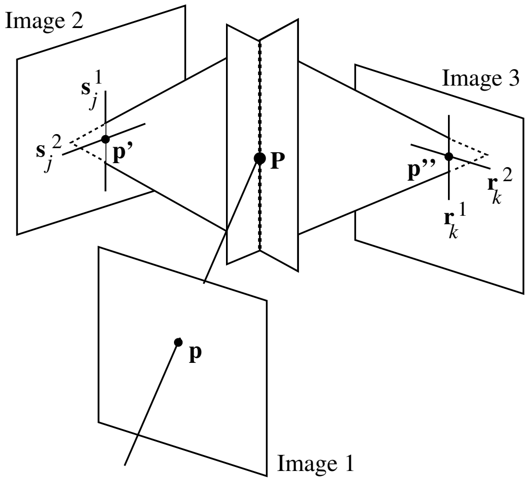


Figure 10: Interpretation of the trilinear tensor (adapted from [2]). $\mathbf{P}$ is the 3-D point with $\mathbf{p}$, $\mathbf{p}'$, and $\mathbf{p}''$ being the projected image points of $\mathbf{P}$.

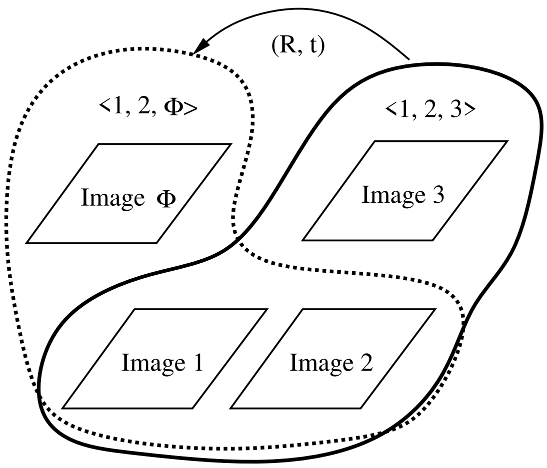


Figure 11: Scheme for generating a novel view (adapted from [2]). Images 1, 2, and 3 are the reference images while image Φ is the novel image to be synthesized. $(R, \mathbf{t})$ is the camera transformation from view 3 to view Φ.

We have implemented a rendering technique based on the trilinear tensor using two reference images. In contrast with Avidan and Shashua's approach, where they assume that the camera focal length is known, we estimate the focal lengths using Kruppa's constraints [23, 15]. The results are shown in Figure 12.

Why would one use the trilinear tensor instead of the fundamental matrix? For one, the trilinear tensor is more stable in comparison to the fundamental matrix. The epipolar geometry becomes unstable under certain camera configurations, such as when the virtual camera location is collinear with the two reference camera locations. In addition, as mentioned earlier, for methods that use fundamental matrices, specification of a new view is through a select number of corresponding points. Specifying a new view using the trilinear tensor is more direct.

6 Discussion

It is interesting to note that image-based rendering has evolved to be an area that straddles both computer vision and computer graphics. This is evident from the range of work in image-based rendering surveyed in this technical report. The notions of the fundamental matrix and trilinear tensor came from computer vision research while ideas related to image warping and resampling are heavily referred to in image processing and computer graphics.

In this survey, image-based rendering techniques are subdivided based primarily on the nature of the pixel indexing scheme, i.e., whether the indexing scheme has a valid 3-D geometric interpretation, uses high dimensional indexing (4-D indexing as in interpolation from dense samples), or uses direct 2-D indexing (as in mosaicking). Division of all these techniques along other axes of discrimination is possible, such as the categories of off-line synthesis (mosaicking), on-line synthesis (geometrically-valid pixel reprojection), and image interpolation (all other techniques) [2].

The commercial applications of image-based rendering have thus far been restricted to viewing of mosaicked wide images. There were enhancements to just pure viewing: faster perspective correction and rendering algorithms, zooming capabilities, jumping to another panoramic location through a predefined "hot-spot" in the current panoramic location, and the addition of multimedia features. However, viewing is still mostly restricted to be from the projection center of the image. It is very likely that it is just a matter of time before other image-based rendering techniques, especially the geometrically-valid pixel reprojection techniques, hit the commercial market.

The challenges that need to be addressed in order to make image-based rendering techniques more useful as a visualization technique include: dealing with visualization of sets of contiguous

Figure 12: Example of visualizing using the trilinear tensor: The top two images are the reference images, with the rest synthesized at arbitrary viewpoints.

wide areas and large sets of reference images, continuous and seamless navigation along a long path across wide areas, and robustly accounting for object or scene occlusions and disocclusions without direct knowledge of 3-D information (which can be hard to obtain for real scenes). Many of the current techniques are either impractical due to their limited viewing scope or too slow due to their high computational requirement. Is it just a case of waiting for processor speeds and I/O bandwidth to increase for certain image-based rendering techniques to be viable? Is it possible to design an "image-based rendering hardware accelerator" that is analogous to the 3-D graphics accelerator?

There is some promise in using image-based rendering techniques for visualization across the Internet and for web applications. One possible use is in background representation or customization for videoconferencing. Only a few images would need to be transmitted initially; processing is done at the client end to update the background scene as the viewpoint changes. One can also envision installing an image-based rendering engine at the server side that provides computed virtual views to the remote client side (say of historical places for a virtual museum website or of houses for real estate businesses—the possibilities are practically endless). The bottlenecks for such an endeavour are bandwidth and processing speed, but these are likely to become less of a factor in the future.

7 Concluding remarks

Despite the promise of the image-based rendering approach, it is not likely going to supplant the 3-D model-based approach. On one hand, the image-based rendering approach does not require 3-D models, sophisticated graphics software and 3-D graphics hardware accelerators, has the property that the execution time is independent of the scene visual complexity, and is capable of very high quality rendering. On the other hand, it may require high processing power, long training time or very large memory requirements, and is inefficient in generating images with very wide fields of view. In addition, sophisticated techniques may have to be incorporated in order to account for object occlusions and disocclusions. It will also be very likely not the approach of choice if goals other than visualization are important, such as spatial reasoning about the scene or object of display, incorporating photometric effects (such as changing highlights and intensity at different camera viewpoints as well as different lighting conditions), and spatial editing (such as adding or removing objects). In these cases, knowledge is required of the object or scene geometry and surface properties. While this kind of information may be extracted from multiple images using nontrivial computer vision techniques (specifically stereo and physics-based vision), there still remain lingering questions such as what is considered to be sufficient in the number of images, the conditions under which the images should be taken, and issues of algorithmic robustness.

More importantly, 3-D model-based rendering is required to produce images of 3-D objects and scenes that do not exist in reality. Typical applications that illustrate this requirement are the visualization of new products designed with the aid of a CAD system, and the production of scenery for sophisticated electronic games based on fantasy.

An optimum rendering system is likely going to employ a hybrid of 3-D model and image-based rendering techniques, or is intelligent enough to decide which type of rendering technique to use for a given set of inputs and goals. As an example, to generate fast rendering of a scene under different imaging conditions, a collection, or *design gallery* [30], of virtual scene views under a range of different imaging conditions at strategically discrete viewpoints may be chosen and computed off-line. These viewpoints may be computed using a 3-D modeler. While the system is on-line, image-based rendering techniques will be used to produce a seamless sequence of images based on user direction. Another promising example is Talisman's idea of subimage reuse for fast and convincing rendering of virtual views [46].

Acknowledgments

We would like to thank Pavan Desikan and Dave DiFranco for their help in writing some image-based rendering code that produced some of the results shown in this technical report. In particular, Pavan extended our preliminary work on image interpolation techniques using multiple cylindrical panoramic images. Dave wrote the program for view synthesis using two uncalibrated images using Avidan's idea of trilinear tensor [2]. We extended the idea by recovering the focal length automatically prior to using the trilinear tensor. We would also like to thank Peter Kochevar and Jim Rehg for their constructive comments that helped improve this technical report, and Michael Jones for kindly supplying Figure 3.

References

[1] E. H. Adelson and J. R. Bergen. The plenoptic function and the elements of early vision. In M. Landy and J. A. Movshon, editors, *Computational Models of Visual Processing*, chapter 1. MIT Press, Cambridge, MA, 1991.

[2] S. Avidan and A. Shashua. Novel view synthesis in tensor space. In *Conference on Computer Vision and Pattern Recognition*, pages 1034–1040, San Juan, Puerto Rico, June 1997.

[3] T. Beier and S. Neely. Feature-based image metamorphosis. *Computer Graphics (SIGGRAPH'92)*, 26(2):35–42, July 1992.

[4] J. F. Blinn. Simulation of wrinkled surfaces. *Computer Graphics (SIGGRAPH'78)*, 12(3):286–292, August 1978.

[5] R. C. Bolles, H. H. Baker, and D. H. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *International Journal of Computer Vision*, 1:7–55, 1987.

[6] P. J. Burt and E. H. Adelson. A multiresolution spline with applications to image mosaics. *ACM Transactions on Graphics*, 2(4):217–236, October 1983.

[7] N. L. Chang and A. Zakhor. View generation for three-dimensional scenes from video sequences. *IEEE Transactions on Image Processing*, 6(4):584–598, April 1997.

[8] S. E. Chen and L. Williams. View interpolation for image synthesis. *Computer Graphics (SIGGRAPH'93)*, pages 279–288, July 1993.

[9] S.E. Chen. QuickTime VR – An image-based approach to virtual environment navigation. *Computer Graphics (SIGGRAPH'95)*, pages 29–38, Aug. 1995.

[10] O. Faugeras. *Three-dimensional computer vision: A geometric viewpoint*. MIT Press, Cambridge, Massachusetts, 1993.

[11] O. D. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In *Second European Conference on Computer Vision (ECCV'92)*, pages 563–578, Santa Margherita Liguere, Italy, May 1992. Springer-Verlag.

[12] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The lumigraph. *Computer Graphics (SIGGRAPH'96)*, pages 43–54, August 1996.

[13] H. Gouraud. Continuous shading of curved surfaces. *IEEE Transactions on Computers*, 20(6):623–628, June 1971.

[14] R. Hartley. In defence of the 8-point algorithm. In *Fifth International Conference on Computer Vision (ICCV'95)*, pages 1064–1070, Cambridge, Massachusetts, June 1995. IEEE Computer Society Press.

[15] R. I. Hartley. Kruppa's equations derived from the fundamental matrix. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):133–135, February 1997.

[16] M. Irani, P. Anandan, and S. Hsu. Mosaic based representations of video sequences and their applications. In *Fifth International Conference on Computer Vision*, pages 605–611, Cambridge, Massachusetts, June 1995.

[17] M. Irani and S. Peleg. Super resolution from image sequences. In *International Conference on Pattern Recognition*, pages 115–120, 1990.

[18] H. Ishiguro, M. Yamamoto, and S. Tsuji. Omni-directional stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):257–262, 1992.

[19] M. Jones. *Multidimensional Morphable Models: A Framework for Representing and Matching Object Classes*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1997.

[20] M. Jones and T. Poggio. Model-based matching of line drawings by linear combination of prototypes. In *Fifth International Conference on Computer Vision*, pages 531–536, Cambridge, Massachusetts, June 1995.

[21] S. B. Kang and R. Szeliski. 3-D scene data recovery using omnidirectional multibaseline stereo. In *Proc.s IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 364–370, June 1996.

[22] A. Katayama, K. Tanaka, T. Oshino, and H. Tamura. A viewpoint dependent stereoscopic display using interpolation of multi-viewpoint images. In S. Fisher, J. Merritt, and B. Bolas, editors, *Stereoscopic Displays and Virutal Reality Systems II, Proc. SPIE*, volume 2409, pages 11–20. 1995.

[23] E. Kruppa. Objektes aus zwei perspektiven mit innerer orientierung. *Sitz.-Ber. Akad. Wiss., Math. Naturw. Kl., Abt. Ila.*, 122:1939–1948, 1913.

[24] C. D. Kuglin and D. C. Hines. The phase correlation image alignment method. In *IEEE 1975 Conference on Cybernetics and Society*, pages 163–165, New York, September 1975.

[25] R. Kumar, P. Anandan, M. Irani, J. Bergen, and K. Hanna. Representation of scenes from collections of images. In *IEEE Workshop on Representations of Visual Scenes*, pages 10–17, Cambridge, Massachusetts, June 1995.

[26] S. Laveau and O. Faugeras. 3-D scene representation as a collection of images and fundamental matrices. Technical Report 2205, INRIA-Sophia Antipolis, February 1994.

[27] M. Levoy and P. Hanrahan. Light field rendering. *Computer Graphics (SIGGRAPH'96)*, pages 31–42, August 1996.

[28] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981.

[29] W. Lorensen and H. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *Computer Graphics (SIGGRAPH'92)*, 21(4):163–169, July 1987.

[30] J. Marks, B. Andalman, P. A. Beardsley, W. Freeman, S. Gibson, J. Hodgins, T. Kang, B. Mirtich, H. Pfister, W. Ruml, K. Ryall, J. Seims, and S. Shieber. Design galleries: A general approach to setting parameters for computer graphics and animation. *Computer Graphics (SIGGRAPH'97)*, pages 389–400, August 1997.

[31] L. McMillan and G. Bishop. Plenoptic modeling: An image-based rendering system. *Computer Graphics (SIGGRAPH'95)*, pages 39–46, August 1995.

[32] D. L. Milgram. Computer methods for creating photomosaics. *IEEE Transactions on Computers*, C(24):1113–1119, 1975.

[33] D. L. Milgram. Adaptive techniques for photomosaicking. *IEEE Transactions on Computers*, C(26):1175–1180, 1977.

[34] V. Nalwa. A true omnidirectional viewer. Technical report, Bell Laboratories, February 1996.

[35] S. Nayar. Catadioptric omnidirectional camera. In *Conference on Computer Vision and Pattern Recognition*, pages 482–488, San Juan, Puerto Rico, June 1997.

[36] S. Peleg. Elimination of seams from photomosaics. *Computer Graphics and Image Processing*, C(16):90–94, 1981.

[37] S. Peleg and J. Herman. Panoramic mosaics by manifold projection. In *Conference on Computer Vision and Pattern Recognition*, pages 338–343, San Juan, Puerto Rico, June 1997.

[38] B. Phong. Illumination for computer-generated pictures. *Communications of the ACM*, 18(6):311–317, June 1975.

[39] H. S. Sawhney and S. Ayer. Compact representations of videos through dominant and multiple motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):814–830, August 1996.

[40] S. M. Seitz and C. R. Dyer. View morphing. *Computer Graphics (SIGGRAPH'96)*, pages 21–30, August 1996.

[41] A. Shashua. Algebraic functions for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):779–789, 1995.

[42] R. Szeliski. Video mosaics for virtual environments. *IEEE Computer Graphics and Applications*, pages 22–30, March 1996.

[43] R. Szeliski and J. Coughlan. Hierarchical spline-based image registration. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'94)*, pages 194–201, Seattle, Washington, June 1994. IEEE Computer Society.

[44] R. Szeliski and S. B. Kang. Direct methods for visual scene reconstruction. In *IEEE Workshop on Representations of Visual Scenes*, pages 26–33, Cambridge, Massachusetts, June 1995.

[45] R. Szeliski and H.-Y. Shum. Creating full view panoramic image mosaics and environment maps. *Computer Graphics (SIGGRAPH'97)*, pages 251–258, August 1997.

[46] J. Torborg and J. T. Kajiya. Talisman: Commodity realtime 3D graphics for the PC. *Computer Graphics (SIGGRAPH'96)*, pages 353–363, August 1996.

[47] T. Vetter and T. Poggio. Linear object classes and image synthesis from a single example image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):733–742, July 1997.

[48] J. Y. A. Wang and E. H. Adelson. Representing moving images with layers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(5):625–638, September 1994.

[49] A. Watt. *3D Computer Graphics*. Addison Wesley, 1993.

[50] G. Wolberg. *Digital Image Warping*. IEEE Computer Society Press, Los Alamitos, California, 1990.

[51] Y. Xiong and K. Turkowski. Creating image-based VR using a self-calibrating fisheye lens. In *Conference on Computer Vision and Pattern Recognition*, pages 237–243, San Juan, Puerto Rico, June 1997.

[52] Y. Yagi and S. Kawato. Panorama scene analysis with conic projection. In *Proceedings of IEEE International Workshop on Intelligent Robots and Systems*, pages 181–187, July 1990.

[53] K. Yamazawa, Y. Yagi, and M. Yachida. Omnidirectional imaging with hyperboloidal projection. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1029–1034, July 1993.

[54] J. Y. Zheng and S. Tsuji. Panoramic representation for route recognition by a mobile robot. *International Journal of Computer Vision*, 9(1):55–76, 1992.

[55] I. Zoghlami, O. Faugeras, and R. Deriche. Using geometric corners to build a 2D mosaic from a set of images. In *Conference on Computer Vision and Pattern Recognition*, pages 420–425, San Juan, Puerto Rico, June 1997.

A Survey of Image-based Rendering Techniques

Sing Bing Kang

CRL 97/4

August 1997