Image-Based Rendering using Image-Warping – Motivation and Background

Leonard McMillan
LCS Computer Graphics Group
MIT

The field of three-dimensional computer graphics has long focused on the problem of synthesizing images from geometric models. These geometric models, in combination with surface descriptions characterizing the reflective properties of each element, represent the scene that is to be rendered. Computationally, the classical computer-graphics image synthesis process is a simulation problem in which light's interactions with the supplied scene description are computed.

Conventional computer vision considers the opposite problem of generating geometric models from images. In addition to images, computer-vision systems depend on accurate camera models and estimates of a camera's position and orientation in order to synthesize the desired geometric models. Often a simplified surface reflectance model is assumed as part of the computer vision algorithm.

The efforts of computer graphics and computer vision are generally perceived as complementary because the results of one field can frequently serve as an input to the other. Computer graphics often looks to the field of computer vision for the generation of complex geometric models, whereas computer vision relies on computer graphics for viewing results. Three-dimensional geometry has been the fundamental interface between the fields of computer vision and computer graphics since their inception. Only recently has this interface come under question. To a large extent the field of *image-based rendering* suggests an alternative interface between the image analysis of computer vision and the image synthesis of computer graphics. In this course I will describe one class of methods for synthesizing images, comparable to those produced by conventional three-dimensional computer graphics methods, directly from other images without an explicit three-dimensional geometric representation.

Another motivation for the development of image-based rendering techniques is that, while geometry-based rendering technology has made significant strides towards achieving photorealism, the process of creating accurate models is still nearly as difficult today as it was twenty-five years ago. Technological advances in three-dimensional scanning methods provide some promise for simplifying the process of model building. However, these automated model acquisition methods also verify our worst suspicions—the geometry of the real world is exceedingly complex. Ironically, one of the primary subjective measures of image quality used in geometry-based computer graphics is the degree to which a rendered image is indistinguishable from a photograph. Consider, though, the advantages of using photographs (images) as the underlying scene representation. Photographs, unlike geometric models, are both plentiful and easily acquired, and, needless to say photorealistic. Images are capable of representing both the geometric complexity and photometric realism of a scene in a way that is currently beyond our modeling capabilities.

Throughout the three-dimensional computer graphics community, researchers, users, and hardware developers alike, have realized the significant advantages of incorporating images, in the form of texture maps, into traditional three-dimensional models. Texture maps are commonly used to add fine surface property variations as well as to substitute for small-scale geometric details. Texture mapping can rightfully be viewed as the precursor to image-based computer graphics methods. In fact, the image-based approach that I present can be viewed as an extension of texture-mapping algorithms commonly used today. However, unlike a purely image-based approach, an underlying three-dimensional model still plays a crucial role with traditional texture maps.

In order to define an image-based computer graphics method, we need a principled process for transforming a finite set of known images, which I will henceforth refer to as reference images, into new images as they would be seen from arbitrary viewpoints. I will call these synthesized images, desired images. Techniques for deriving new images based on a series of reference images or drawings are not new. A skilled architect, artist, draftsman, or illustrator can, with relative ease, generate accurate new renderings of an object based on surprisingly few reference images. These reference images are frequently illustrations made from certain cardinal views, but it is not uncommon for them to be actual photographs of the desired scene taken from a different viewpoint. One characterization of image-based rendering is to emulate the finely honed skills of these artisans by using computational powers.

While image-based computer graphics has come many centuries after the discovery of perspective illustration techniques by artists, its history is still nearly as long as that of geometry-based computer graphics. Progress in the field of image-based computer graphics can be traced through at least three different scientific disciplines. In photogrammetry the problems of distortion correction, image registration, and photometrics have progressed toward the synthesis of desired images through the composition of reference images. Likewise, in computer vision, problems such as navigation, discrimination, and image understanding have naturally led in the same direction. In computer graphics, as discussed previously, the progression toward image-based rendering systems was initially motivated by the desire to increase the visual realism of the approximate geometric descriptions. Most recently, methods have been introduced in which the images alone constitute the overall scene description. The remainder of this introduction discusses previous works in image-based computer graphics and their relationship to the image-warping approach that I will present.

In recent years, images have supplemented the image generation process in several different capacities. Images have been used to represent approximations of the geometric contents of a scene. Collections of images have been employed as databases from which views of a desired environment are queried. And, most recently, images have been employed as full-fledged scene models from which desired views are synthesized. In this section, I will give an overview of the previous work in image-based computer graphics partitioned along these three lines.

Images, mapped onto simplified geometry, are often used as an approximate representation of visual environments. Texture mapping is perhaps the most obvious example of this use. Another more subtle approximation involves the assumption that all, or most, of the geometric content of a scene is located so far away from the viewer that its actual shape is inconsequential. Much of the pioneering work in texture mapping is attributable to the classic work of Catmull, Blinn, and Newell. The flexibility of image textures as three-dimensional computer graphics primitives has since been extended to include small perturbations in surface orientation (bump maps) [Blinn76] and approximations to global illumination (environment and shadow mapping) [Blinn76] [Greene86] [Segal92]. Recent developments in texture mapping have concentrated on the use of visually rich textures mapped onto very approximate geometric descriptions [Shade96] [Aliaga96][Schaufler96].

Texture mapping techniques rely on mapping functions to specify the relationship of the texture's image-space coordinates to their corresponding position on a three-dimensional model. A comprehensive discussion of these mapping techniques was undertaken in [Heckbert86]. In practice the specification of this mapping is both difficult and time consuming, and often requires considerable human intervention. As a result, the most commonly used mapping methods are restricted to very simple geometric descriptions, such as polygonal facets, spheres and cylinders.

During the rendering process, these texture-to-model mapping functions undergo another mapping associated with the perspective-projection process. This second mapping is from the three-dimensional space of the scene's representation to the coordinate space of the desired image. In actual rendering systems, one or both of these mapping processes occurs in the opposite or inverse order. For instance, when ray tracing, the mapping of the desired image's coordinates to the three-dimensional coordinates in the space of the visible object occurs first. Then, the mapping from the three-dimensional object's coordinates to the texture's image-space coordinates is found. Likewise, in z-buffering based methods, the mapping from the image-space coordinate to texture's image-space occurs during the rasterization process. These inverse methods are known to be subject to aliasing and reconstruction artifacts. Many techniques, including mip-maps [Williams83] and summed-area tables [Crow84] [Glassner86], have been suggested to address these problems.

Another fundamental limitation of texture maps is that they rely solely on the geometry of the underlying three-dimensional model to specify the object's shape. The precise representation of three-dimensional shape using primitives suitable for the traditional approach to computer graphics is, in itself, a difficult problem that has long been an active topic in computer graphics research. When the difficulties of representing three-dimensional shape are combined with the issues of associating a texture coordinate to each point on the surface (not to mention the difficulties of acquiring suitable textures in the first place), the problem becomes even more difficult. It is conceivable that, given a series of photographs, a three-dimensional computer model could be assembled. And, from those same photographs, various figures might be identified, cropped, and the perspective distortions removed so that a texture might be extracted. Then, using traditional three-dimensional computer graphics methods, renderings of any desired image could be computed. While the process outlined is credible, it is both tedious and prone to errors. The image-based approach to computer graphics described in this thesis attempts to sidestep many of these intermediate steps by defining mapping functions from the image-space of one or more reference images directly to the image-space of a desired image.

A new class of scene approximation results when an image is mapped onto the set of points at infinity. The mapping is accomplished in exactly the same way that texture maps are applied to spheres, since each point on a sphere can be directly associated with another point located an infinite distance from the sphere's center. This observation is also the basis of environment maps. However, environment maps are observed indirectly as either reflections within other objects or as representations of a scene's illumination environment. When such an image mapping is intended for direct viewing, a new type of scene representation results. An image-based computer graphics system of this type, called QuickTimeVR [Chen95], has been developed by Apple Computer. In QuickTimeVR, the underlying scene is represented by a set of cylindrical images. The system is able to synthesize new planar views in response to a user's input by warping one of these cylindrical images. This is accomplished at highly interactive rates (greater than 20 frames per second) and is done entirely in software. The system adapts both the resolution and reconstruction filter quality based on the rate of the interaction. QuickTimeVR must be credited with exposing to a wide audience the vast potential of image-based computer graphics. The QuickTimeVR system is a reprojection method. It is only capable of describing image variations due to changes in viewing orientation. Translations of the viewing position can only be approximated by selecting the cylindrical image whose center-of-projection is closest to the current viewing position.

The panoramic representation afforded by the cylindrical image description provides many practical advantages. It immerses the user within the visual environment, and it eliminates the need to consider the viewing angle when determining which reference image is closest to a desired view. However, several normal photographs are required to create a single cylindrical projection. These images must be properly registered and then reprojected to construct the cylindrical reference image. QuickTimeVR's image-based approach has significant similarity to the approach described here. Its rendering process is a special case of the cylinder-to-plane warping equation in the case where all image points are computed as if they were an infinite distance from the observer.

The movie-map system by Lippman [Lippman80] was one of the earliest attempts at constructing a purely image-based computer graphics system. In a movie-map, many thousands of reference images were stored on interactive video laser disks. These images could be accessed randomly, according to the current viewpoint of the user. The system could also accommodate simple panning, tilting, or zooming about these fixed viewing positions. The movie-map approach to image-based computer graphics can also be interpreted as a table-based approach, where the rendering process is replaced by a database query into a vast set of reference images. This database-like structure is common to other image-based computer graphics systems. Movie-maps were unable to reconstruct all possible desired views. Even with the vast storage capacities currently available on media such as laser disks, and the rapid development of even higher capacity storage media, the space of all possible desired images appears so large that any purely database-oriented approach will continue to be impractical in the near future. Also, the very subtle differences between images observed from nearby points under similar viewing conditions bring into question the overall efficiency of this approach. The image-based rendering approach described here could be viewed as a reasonable compression method for movie maps.

Levoy and Hanrahan [Levoy96] have developed another database approach to image-based computer graphics in which the underlying modeling primitives are rays rather than images. A key innovation of this technique, called light-field rendering, is the recognition that all of the rays that pass through a slab of empty space enclosed between two planes can be described using only four parameters rather than the five dimensions required for the typical specifications of a ray. They also describe an efficient technique for generating the ray parameters needed to construct any arbitrary view. The subset of rays originating from a single point on a light field's entrance plane can be considered as an image corresponding to what would have been seen at that point. The entire two-parameter family of images originating from points on the entrance plane can be considered as a set of reference images. During the rendering process, the three-dimensional entrance and exit planes are projected onto the desired viewing plane. The final image is constructed by determining the image-space coordinates of the two points visible at a specified pixel coordinate one coordinate from the projected image of the entrance plane and the second from the image of the exiting plane). The desired ray can be looked up in the light field's database of rays using these four parameter values. Image generation using light fields is inherently a database query process, much like the movie map image-based process. The storage requirements for a lightfield's database of rays can be very large. Levoy and Hanrahan discuss a lossy method for compressing light fields that attempts to minimize some of the redundancy in the light-field representation.

The lumigraph [Gortler96] is another ray-database query algorithm closely related to the light-field. It also uses a four-dimensional parameterization of the rays passing through a pair of planes with fixed orientations. The primary differences in the two algorithms are the acquisition methods used and the final reconstruction process. The lumigraph, unlike the light field, considers the geometry of the underlying models when reconstructing desired views. This geometric information is derived from image segmentations based on the silhouettes of image features. The preparation of the ray database represented in a lumigraph requires considerable preprocessing when compared to the light field. This is a result of the arbitrary camera poses that are used to construct the database of visible rays. In a lightfield, though, the reference images are acquired by scanning a camera along a plane using a motion platform. The lumigraph reconstruction process involves projecting each of the reference images as they would have appeared when mapped onto the exit plane. The exit plane is then viewed through an aperture on the entrance plane surrounding the center-of-projection of the reference image. When both the image of the aperture on the entrance plane and the reference image on the exit plane are projected as they would be seen from the desired view, the region of the reference image visible through the aperture can be drawn into the desired image. The process is repeated for each reference image. The lumigraph's approach to image-based three-dimensional graphics uses geometric information to control the blending of the image fragments visible through these apertures. Like the lightfield, the lumigraph is a data intensive rendering process.

The image-warping approach to IBR discussed here attempts to reconstruct desired views based on far less information. First, the reference image nearest the desired view is used to compute as much of the desired view as possible. Regions of the desired image that cannot be reconstructed based on the original reference image are subsequently filled in from other reference images. The warping approach to IBR can also be considered as a compression method for both light fields and lumigraphs. Considering the projective constraints induced by small variations in the viewing configuration reduces redundancy of the database representation. Thus, an image point, along with its associated mapping function, can be used to represent rays in many different images from which the same point is visible.

Many other computer graphics methods have been developed where images serve as the underlying representation. These methods handle the geometric relationships between image points very differently. In the case of image morphing, the appearance of a dynamic geometry is often a desired effect. Another method, known as *view interpolation* relies on an approximation to a true projective treatment in order to compute the mapping from reference images to desired images. Also, additional geometric information is required to determine correct visibility. A third method, proposed by Laveau and Faugeras, is based on an entirely projective approach to image synthesis. However, they have chosen to make a far more restrictive set of assumptions in their model, which allows for an ambiguous Euclidean interpretation.

Image morphing is a popular image-based computer graphics technique [Beier92], [Sietz96], [Wolberg90]. Generally, morphing describes a series of images representing a transition between two reference images. These reference images can be considered as endpoints along some path through time and/or space. An interpolation process is used to reconstruct intermediate images along the path's trajectory. Image morphing techniques have been used to approximate dynamic changes in camera pose [Sietz96], dynamic changes in scene geometry [Wolberg90], and combinations of these effects. In addition to reference images, the morphing process requires that some number of points in each reference be associated with corresponding points in the other. This association of points between images is called a *correspondence*. This extra information is usually hand crafted by an animator.

Most image-morphing techniques make the assumption that the transition between these corresponding points occurs at a constant rate along the entire path, thus amounting to a linear approximation. Also, a graduated blending function is often used to combine the reference images after they are mapped from their initial configuration to the desired point on the path. This blending function is usually some linear combination of the two images based on what percentage of the path's length has been traversed. The flexibility of image-morphing methods, combined with the fluidity and realism of the image transitions generated, have made a dramatic impact on the field of computer graphics, especially when considering how recently they have been developed. A subset of image morphing, called view morphing, is a special case of image-based computer graphics. In view morphing the scene geometry is fixed, and the pose of the desired views lies on a locus connecting the centers-of-projection of the reference images. With the notable exception of the work done by Seitz, general image morphing makes no attempt to constrain the trajectory of this locus, the characteristics of the viewing configurations, or the shapes of the objects represented in the reference images. In this thesis, I will propose image-mapping functions that will allow desired images to be specified from any viewing point, including those off the locus. Furthermore, these mapping functions, like those of Sietz, are subject to constraints that are consistent with prescribed viewing conditions and the static Euclidean shape of the objects represented in the reference images.

Chen and Williams [Chen93] have presented a view interpolation method for three-dimensional computer graphics. It uses several reference images along with image correspondence information to reconstruct desired views. Dense correspondence between the pixels in reference images is established by a geometry-based rendering preprocess. During the reconstruction process, linear interpolation between corresponding points is used to map the reference images to the desired viewpoints, as in image morphing. In general, this interpolation scheme gives a reasonable approximation to an exact reprojection as long as the change in viewing position is slight. Indeed, as the authors point out, in some viewing configurations this interpolation is exact. Chen and Williams acknowledge, and provide a solution for, one of the key problems of image-based rendering—visible surface determination. Chen and Williams presort a quadtree-compressed flow-field in a back-to-front order according to the scene's depth values. This approach works only when all of the partial sample images share a common gaze direction and the synthesized viewpoints are restricted to stay within 90 degrees of this gaze angle. The underlying problem is that correspondence information alone (i.e., without depth values) still allows for many ambiguous visibility solutions unless we restrict ourselves to special flow fields that cannot fold (such as rubber-sheet local spline warps or thin-plate global spline warps).

Establishing the dense correspondence information required for a view interpolation system can also be problematic. Using pre-rendered synthetic images, Chen and Williams were able to determine the association of points by using the depth values stored in a z-buffer. In the absence of a geometric model, they suggest that approximate correspondence information can be established for all points using correlation methods. The image-based approach to three-dimensional computer graphics described in this research has a great deal in common with the view interpolation method. For instance, both methods require dense correspondence information in order to generate the desired image, and both methods define image-space to image-space mapping functions. In the case of view interpolation, the correspondence information is established on a pairwise basis between reference images. As a result the storage requirements for the correspondence data associating N reference images is O(N). My approach is able to decouple the correspondence information from the difference in viewing geometries. This allows a single flow field to be associated with each image, requiring only O(N) storage. Furthermore, the approach to visibility used in my method does not rely on any auxiliary geometric information, such as the presorted image regions based on the z-values, used in view interpolation.

Laveau's and Faugeras' [Laveau94] image-based computer-graphics system takes advantage of many recent results from computer vision. They consider how a particular projective geometric structure called epipolar geometry can be used to constrain the potential reprojections of a reference image. They explain how a fundamental matrix can describe the projective shape of a scene with scalar values defined at each image point. They also provide a two-dimensional ray-tracing-like solution to the visibility problem that does not require an underlying geometric description. Yet, it might require several images to assure an unambiguous visibility solution. Laveau and Faugeras also discuss combining information from several views, though primarily for the purpose of resolving visibility as mentioned before. By relating the reference views and the desired views by the homogenous transformations between their projections, Laveau and Faugeras can compute exact perspective depth solutions. However, the solutions generated using Laveau and Faugeras' techniques do not reflect an unambiguous Euclidean environment. Their solution is consistent with an entire family of affine coordinate frames. They have pointed out elsewhere [Faugeraus92b] that when additional constraints are applied, such as the addition of more reference images and the requirement that a fixed camera model be used for all references images, then a unique Euclidean solution can be assured.

The methods described by Laveau and Faugeras are very similar to the image-based approach to computer graphics described here. The major difference in my approach is that I assume that more information is available than simply the epipolar geometries between reference images. Other significant differences are that the forward-mapping approach described here has fewer restrictions on the desired viewing position, and I provide a simpler solution to the visibility problem.

The delineation between computer graphics and computer vision has always been at the point of a geometric description. In IBR we tend to draw the lines somewhat differently. Rather than beginning the image synthesis task with a geometric description, we begin with images. In this overview I presented a summary of the various image-based rendering methods frequently used today.

Images have already begun to take on increasingly significant roles in the field of computer graphics. They have been successfully used to enhance the apparent visual complexity of relatively simple geometric screen descriptions. The discussion of previous work showed how the role of images has progressed from approximations, to databases, to actual scene descriptions.

In this course I will present an approach to three-dimensional computer graphics in which the underlying scene representation is composed of a set of reference images. I will present algorithms that describe the mapping of image-space points in these reference images to their image-space coordinates in any desired image. All information concerning the three-dimensional shape of the objects seen in the reference images will be represented implicitly using a scalar value that is defined for each point on the reference image plane. This value can be established using point correspondences between reference images. I will also present an approach for computing this mapping from image-space to image-space with correct visibility, independent of the scene’s geometry.