Shape as a Perturbation to Projective Mapping

Leonard McMillan and Gary Bishop

Department of Computer Science
University of North Carolina,
Sitterson Hall, Chapel Hill, NC 27599
email: mcmillan@cs.unc.edu gb@cs.unc.edu

1.0 Introduction

In the classical computer-graphics approach to three-dimensional rendering, a scene is described in terms of its geometry and surface properties. Given this representation, the rendering task can be thought of as a physical simulation problem. Recently, there has been an increased interest in alternate rendering paradigms, most of which fall into a category that we call *image-based* rendering. In these alternate approaches, reference images, rather than geometry and surface properties, are used as the primary scene description. In this paper we describe a simple extension to the well known results of projective geometry, resulting in an efficient foundation for the construction of image-based rendering systems.

The fact that all planar projections of a given three-dimensional planar surface are related by two-dimensional homogeneous transforms, known as perspective mappings, is an important result of projective geometry. While the three-dimensional projection process defines mappings from $R^3 \rightarrow R^2$, these two-dimensional homogeneous relationships define transformations from $R^2 \rightarrow R^2$, where the domains and ranges are in the images' coordinate systems. These perspective mappings can be easily expressed in matrix form as shown below:

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

where

$$x' = \frac{u}{w} \qquad \text{and} \qquad y' = \frac{v}{w}$$

By setting g and h to zero, we can see that two-dimensional affine mappings are a proper subset of the perspective case. Mann and Picard [Mann94] have also shown how all planar projections about a common center of projection are related by perspective mappings.

When perspective mappings are considered as image-warping functions [Heckbert89] [Wolberg90], several practical benefits are exposed. First, since perspective mappings are of rank three with eight degrees of freedom, they are uniquely invertible. This allows the image warping process to proceed in either a forward-mapping or inverse-mapping fashion[1]. Second, the linear form of the denominators and numerators of the mapping functions makes them well suited for incremental evaluation. The combination of these two factors explains the underlying computational advantage of texture mapping over pure geometric representation in computer graphics. However, the reason why textures play only a supplemental role in traditional three-dimensional image generation is that a single perspective transform can only convey planar shapes. Thus, they are generally considered as merely an augmentation to the shading process rather than a shape description.

In the following derivation we will show how perspective mapping, with a simple modification, can represent shape as well as changes of viewing position. When evaluated as a two-dimensional image warp, this modified mapping maintains most of the advantages of the original method.

2.0 Derivation

We use a series of backprojections and reprojections to derive the modified perspective mapping function. Throughout, we will use the following model of a three-dimensional projection that retains our image-space orientation. A point and three vectors, ($\dot{p}$, $\vec{o}$, $\vec{u}$, and $\vec{v}$), describe the projection. As shown in Figure 1, the point, $\dot{p}$, determines the center of projection, $\vec{o}$ denotes a vector from the center of projection to the origin of the viewing plane, and $\vec{u}$, $\vec{v}$ form a basis set for spanning the viewing plane. This formulation is capable of describing any planar projection, and it naturally allows for skewed and off-axis projections.

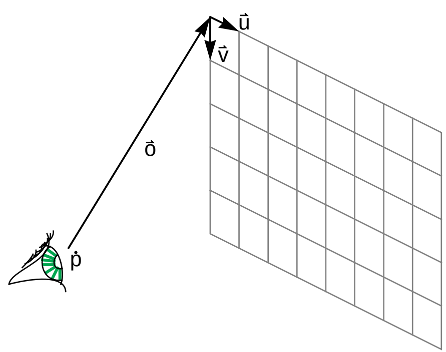

FIGURE 1. The center-of-projection and three vectors determine the planar projection.

Only the relative magnitudes of the vectors $\vec{o}$, $\vec{u}$, and $\vec{v}$ are significant. Thus, it is often convenient to scale them such that one is unit-length. We define a matrix, M, and its

1. A forward-mapping process computes destination pixels by accumulating successive source pixel contributions. Inverse-mapping determines the subset of source pixels that contribute to a given destination pixel.

inverse as follows. M is the concatenation of the column vectors, $\vec{u}$, $\vec{v}$, and $\vec{o}$. The rows of the inverse of M are defined as cross-products of the projection vectors. This formulation of the adjoint matrix provides a useful geometric interpretation.

$$M = \begin{bmatrix} u_X & v_X & o_X \\ u_Y & v_Y & o_Y \\ u_Z & v_Z & o_Z \end{bmatrix} \qquad M^{-1} = \begin{bmatrix} \vec{v} \times \vec{o} \\ \vec{o} \times \vec{u} \\ \vec{u} \times \vec{v} \end{bmatrix} \frac{1}{|M|} = \frac{N}{|M|}$$

The back projection of an image coordinate, *(x, y)*, into three-space is defined as

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \dot{p} + M\hat{x}\frac{r(x, y)}{|M\hat{x}|},$$

where $\hat{x}$ is the two-dimensional homogeneous image coordinate, $\hat{x} = \begin{bmatrix} x & y & 1 \end{bmatrix}^T$, and $\frac{r(x, y)}{|M\hat{x}|}$ is the normalized range function defined for each point on the image plane. The projection of a point in space, *(X, Y, Z)*, onto an image plane is given by the expression,

$$\hat{x} \approx N\left(\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} - \dot{p}\right),$$

where ≈ denotes equivalence down to a scale factor. The combination of these expressions establishes the relationship between images of the same point in two different projections, $P_1\{\dot{p}_1, \vec{o}_1, \vec{u}_1, \vec{v}_1, (x_1, y_1)\}$ and $P_2\{\dot{p}_2, \vec{o}_2, \vec{u}_2, \vec{v}_2, (x_2, y_2)\}$. This relationship is determined by the following expression,

$$\hat{x}_2 \approx N_2\left(\dot{p}_1 + M_1\hat{x}_1\frac{r_1(x_1, y_1)}{|M_1\hat{x}_1|} - \dot{p}_2\right).$$

After multiplying through and reorganizing terms we arrive at

$$\hat{x}_2 \approx N_2(\dot{p}_1 - \dot{p}_2) + N_2M_1\hat{x}_1\frac{r_1(x_1, y_1)}{|M_1\hat{x}_1|}.$$

Next, the second term's dependence on the range function can be factored out by scaling the result by its normalized inverse,

$$\delta(x, y) = \frac{|M\hat{x}|}{r(x, y)},$$

giving

$$\hat{x}_2 \approx N_2(\dot{p}_1 - \dot{p}_2)\,\delta(x_1, y_1) + N_2M_1\hat{x}_1.$$

We call $\delta(x, y)$ the generalized disparity term for reasons that we will discuss later.

We can now specify the mapping functions as the following rational expressions:

$$x_2 = \frac{ax_1 + by_1 + c + k\delta_1(x, y)}{gx_1 + hy_1 + i + m\delta_1(x, y)} \qquad y_2 = \frac{dx_1 + ey_1 + f + l\delta_1(x, y)}{gx_1 + hy_1 + i + m\delta_1(x, y)}$$

where

$$a = \vec{u}_1 \cdot \vec{r} \qquad b = \vec{v}_1 \cdot \vec{r} \qquad c = \vec{o}_1 \cdot \vec{r} \qquad k = (\dot{p}_1 - \dot{p}_2) \cdot \vec{r}$$

$$d = \vec{u}_1 \cdot \vec{s} \qquad e = \vec{v}_1 \cdot \vec{s} \qquad f = \vec{o}_1 \cdot \vec{s} \qquad l = (\dot{p}_1 - \dot{p}_2) \cdot \vec{s}$$

$$g = \vec{u}_1 \cdot \vec{n} \qquad h = \vec{v}_1 \cdot \vec{n} \qquad i = \vec{o}_1 \cdot \vec{n} \qquad m = (\dot{p}_1 - \dot{p}_2) \cdot \vec{n}$$

and

$$\vec{r} = \vec{v}_2 \times \vec{o}_2 \qquad \vec{s} = \vec{o}_2 \times \vec{u}_2 \qquad \vec{n} = \vec{u}_2 \times \vec{v}_2.$$

Compare this formulation to the expression for the planar perspective mapping function.

$$x_2 = \frac{ax_1 + by_1 + c}{gx_1 + hy_1 + i} \qquad y_2 = \frac{dx_1 + ey_1 + f}{gx_1 + hy_1 + i}$$

If the value of the generalized disparity function is defined to be $\delta(x, y) = 0$, which corresponds to $r(x, y) = \infty$, then our formulation matches the perspective mapping case. In fact, any constant-valued disparity function is equivalent to a translation and uniform scaling of the planar perspective mapping at infinity. One can visualize the disparity function as selecting an appropriate slice from a set of parallel planes undergoing a common projection. For this reason, the disparity function is merely a perturbation to perspective mapping.

3.0 Discussion

In our derivation of the perturbed projective mapping function, we defined the generalized disparity value in terms of the image's range function. This is equivalent to considering the image to be a bivariate function with an explicit geometric definition. In this section we will show how an explicit determination of the scene's geometry is unnecessary when determining the image warping function needed to reproject a given image to a new viewing position.

We begin by comparing the generalized disparity value, given previously, to the expression for stereo-disparity in the case of cameras with parallel optical axes [Faugeras93] as given below:

$$\text{disparity} = \frac{\text{baseline} \times \text{focal length}}{\text{depth}}$$

In the stereo-disparity case, the depth at a point on the image is determined by its projection onto a unit vector along the optical axis. This formulation makes sense when both images share a common optical axis that is assumed to be perpendicular to a common image-plane. In determining depth-from-stereo, this geometric orientation arises because it is required for epipolar lines to map to image scan lines. However, this relationship does not hold for arbitrary viewing parameters. Generally, an image rectification[1] step is required to align the optical axis of an arbitrary projection in order to satisfy the condition that the projection of epipolar planes correspond to image scan lines. In contrast, the range function is simply defined as the radial distance from the center of projection to the point. This implies that the generalized disparity function does not assume the existence of any preferred direction, such as the optical axis of the stereo-disparity case.

The stereo-disparity function can be related to the generalized disparity function on a term-by-term basis. The role of the baseline is replaced by the vector, $\dot{p}_1 - \dot{p}_2$, connecting the initial and the desired centers of projection. This is the vector that appears in the computation of the k, l, and m terms of the perturbed projective mapping function. The focal length is related to the generalized disparity function's normalizing term, $|M\hat{x}|$. This normalization factors out of the disparity calculation the distance from the center of projection to each point on the image plane, whereas the focal length corresponds to this distance along the optical axis. Thus, since all points include their own scale factor, they can be treated uniformly, eliminating the special distinction of the optical-axis. Finally, the depth factor of stereo-disparity is related to the range value of the generalized disparity function, as discussed previously.

A similar relationship can be shown between the generalized disparity function and the translational component of the optical-flow function, as derived by [Prazdny83]. This suggests several possibilities. First, consider that it is straightforward to convert stereo-disparity values to the generalized disparity form. This merely involves a scaling of the stereo-disparity values by the secant of the angle between the pixel's ray and the optical axis, and dividing out the length of the baseline. This scalar-valued generalized-disparity function allows us to warp images so that they behave like a three-dimensional model, and it does so using directly observable image measurements, (i.e. disparity values or optical flow) without needing to create an explicit geometric representation.

While the perturbed projective mapping function correctly determines the image coordinates of each pixel in the resulting projection, the possibility exists that it may also introduce many-to-one mappings, called topological folds. A situation is depicted in Figure 2 where two different images result from the same reference image and image-warp functions; the only difference was the order of evaluation. Ideally, only the front-most surface

1. The image rectification step is equivalent to a perspective image mapping.

would be displayed. Determining the correct visible surface at each pixel position is one of the fundamental problems of traditional computer graphics.

One method for determining the correct visibility is to treat the image as a spatial height-field and use traditional computer graphics techniques to transform, project, and scan convert the result. A standard Z-buffer algorithm could be used to determine the visibility on a pixel-by-pixel basis. There are shortcomings with this approach, however. The transformation process requires the computation of an additional rational expression to determine the z-value for each pixel. Also, a screen-sized memory array is required to store these z-values. This approach is nearly identical to a geometric representation[1].

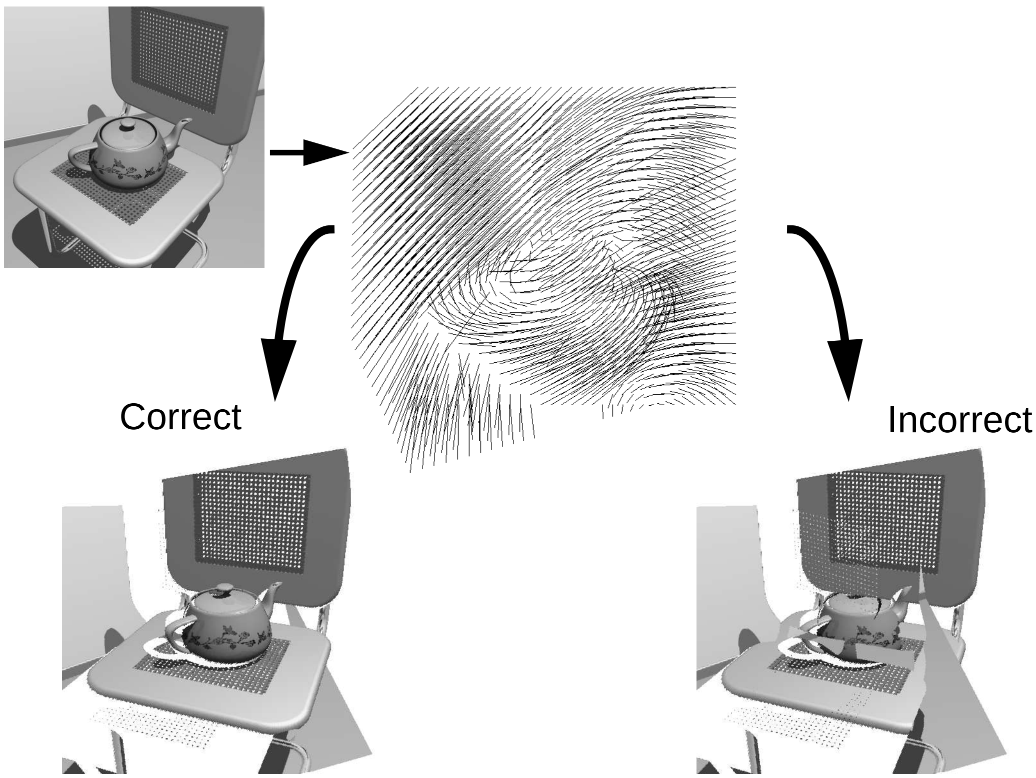

FIGURE 2. The image warp's mapping function allows invalid visibility solutions.

We have developed an alternate approach to determining the visible surface that does not require an explicit conversion to a geometric representation and has the following important properties. It determines a unique evaluation order for computing the image-warp function such that surfaces are drawn in a back-to-front order; thus, it allows a simple painter's style visibility calculation. It maintains the spatial coherence of the image, allowing the warp function to be computed incrementally. And, most significantly, the enumeration order can be computed independent of the generalized disparity function,

1. We say nearly because a clever implementation might take some advantage of the image-height field's spatial coherence.

$\delta(x, y)$. This last property is surprising since all of the information concerning the shape of the underlying data is represented by this disparity information. Our visibility algorithm is capable of determining the visible surface using only the centers-of-projection of the reference and desired images, along with the projection parameters of the reference image. This visibility approach allows us to use only standard image warping methods without any explicit appeal to the geometric content of the scene. A detailed discussion of the algorithm can be found in [McMillan95a] and [McMillan95b].

4.0 Results

We have developed several prototype image-based rendering systems using the methods and algorithms described. Each system was written in C and operates in a standard UNIX environment as an X-windows application. Three of these systems are briefly described below.

We have constructed a general purpose planar image-warping system that takes as input a reference image and either its corresponding disparity or range image. This system is capable of generating arbitrary reprojections of a 512×512 reference image in excess of 10 frames per second. Unlike traditional computer graphics systems, the performance of our image-based rendering system is independent of the geometric complexity of the underlying scene. Instead, the performance is determined by the sizes of the reference image and the reprojected result. Example outputs are shown in Figure 3.

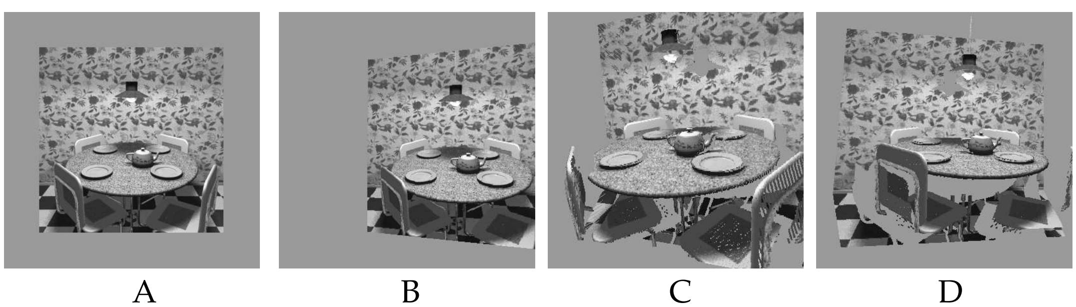

FIGURE 3. Example images output from an image-based rendering system. The input reference image is shown in Image A. Image B shows a perspective mapping. Images C and D show perturbed perspective mappings.

We have also built a head-tracked stereoscopic image-based rendering system, as described in [McMillan95a], to evaluate the effectiveness of the shape generated by the perturbed projective mapping. While the system allows for separate reference images for the left and right eyes, we have found that the illusion of a solid object in three-space can be just as convincing with only a single reference image. The warped left-eye and right-eye images are easily fused stereoscopically even without independent monocular occlusion information. We have also been pleasantly surprised with the size of the useful working range for the warped images generated from a single reference image. Figure 4 shows a tracked user walking around an image-based model synthesized from a single reference image.

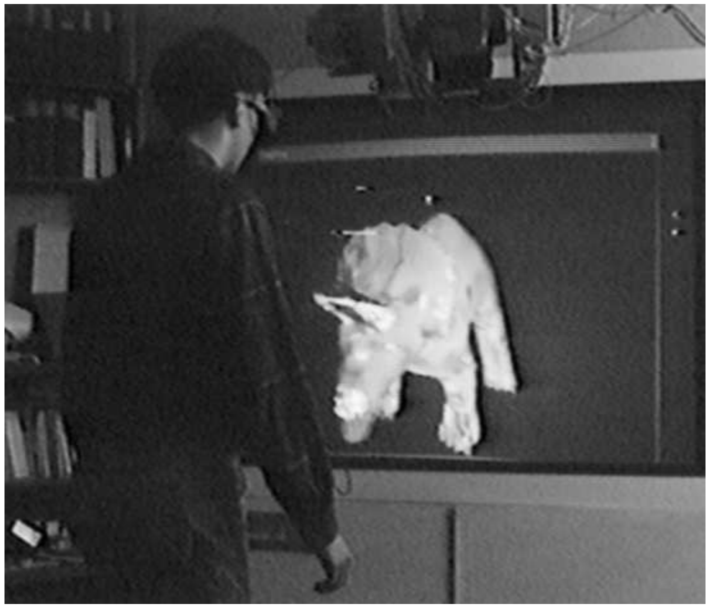

FIGURE 4. A user demonstrating a head-tracked stereoscopic image-rendering system. Both the left eye and right eye images result from warping a single reference image.

The image warping methods presented can be easily extended to handle non-planar projection manifolds. We have developed a third image-based rendering system which uses cylindrical projections as reference images, and generates planar reprojections from arbitrary viewing positions. With this system we hope to generate truly immersive virtual environments. Example outputs from this system are shown in Figure 5.

FIGURE 5. A cylindrical reference image generated from 36 planar projections (top) and three planar reprojections (bottom) each generated from a different center of projection, but, with correct apparent depth and occlusion.

5.0 Conclusions

We have presented a simple perturbation to projective mapping that provides a representation of three-dimensional shape as a scalar function defined on the image's local coordinate system. We have shown how this scalar function, which we call the generalized disparity function, relates to directly observable image features. Since an image-warping function for generating arbitrary reprojections of the scene can be established by knowing only the value of this disparity field and various projection parameters of the reference image and the desired view, we consider this image-based rendering paradigm independent of an explicit geometric model. While the information required to generate such a model is inherent in our representation, we at no time build or process one.

There are considerable efficiencies in using the perturbed projective mapping approach for scene generation. First, the rational-linear expressions describing the arbitrary image warps allow for a simple incremental evaluation in screen space. Second, an image enumeration order which guarantees a correct visibility solution can be determined without the need for a three-dimensional representation, or a screen-sized depth-buffer. And finally, the computation required to compute an image warp is independent of the geometric complexity of the scene represented. The computational requirement is instead determined only by the number of pixels in the reference and desired images.

We have demonstrated how our approach to representing shape can form the basis for an image-based scene representation. The techniques that we describe are well suited to a hardware implementation. In fact, it seems entirely possible that existing texture-mapping hardware could be used to compute our perturbed perspective mappings with little more than a firmware modification.

6.0 References

[Chen93] Chen, S. E. and L. Williams. *"View Interpolation for Image Synthesis,"* **Computer Graphics** (SIGGRAPH'93 Proceedings), pp. 279-288, July 1993.

[Faugeras93] Faugeras, O., **Three-dimensional Computer Vision: A Geometric Viewpoint**, The MIT Press, Cambridge, Massachusetts, 1993.

[Hartley94] Hartley, R.I., *"Self-Calibration from Multiple Views with a Rotating Camera,"* **Proceedings of the European Conference on Computer Vision**, May 1994.

[Heckbert89] Heckbert, P. S., *"Fundamentals of Texture Mapping and Image Warping,"* Masters Thesis, Dept. of EECS, **UCB, Technical Report No. UCB/CSD 89/516**, June 1989.

[Laveau94] Laveau, S. and O. Faugeras, *"3-D Scene Representation as a Collection of Images and Fundamental Matrices,"* **INRIA, Technical Report No. 2205**, February, 1994.

[Mann94] Mann, S. and R. W. Picard, *"Virtual Bellows: Constructing High Quality Stills from Video,"* **Proceedings of the First IEEE International Conference on Image Processing**, November 1994.

[McMillan95a] McMillan, Leonard, and Gary Bishop, *"Head-Tracked Stereo Display Using Image Warping,"* **1995 IS&T/SPIE Symposium on Electronic Imaging Science and Technology,** SPIE Proceedings #2409A, (San Jose, CA) February 5-10, 1995.

[McMillan95b] McMillan, Leonard, *"A List-Priority Rendering Algorithm for Redisplaying Projected Surfaces,"* **UNC Technical Report TR95-005,** University of North Carolina, 1995.

[McMillan95c] McMillan, Leonard, *"Acquiring Immersive Virtual Environments with an Uncalibrated Camera,"* **UNC Technical Report TR95-006,** University of North Carolina, 1995.

[Prazdny83] Prazdny, K., *"On the Information in Optical Flows,"* **Computer Vision, Graphics, and Image Processing**, Vol. 22, No. 9, 1983, pp 239-259.

[Szeliski94] Szeliski, R., *"Image Mosaicing for Tele-Reality Applications,"* **DEC and Cambridge Research Lab Technical Report**, CRL 94/2, May 1994.

[Wolberg90] Wolberg, G., **Digital Image Warping,** IEEE Computer Society Press, Los Alamitos, CA, 1990.