Combining Multiple Reference Images in an Inverse Warper

by

Lawrence W. Chang

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degrees of

Bachelor of Science in Computer Science and Engineering and Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 22, 1998

Author ....
Department of Electrical Engineering and Computer Science
May 22, 1998

Certified by ....
Leonard McMillan
Department of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by ....
Arthur C. Smith
Chairman, Department Committee on Graduate Theses

Combining Multiple Reference Images in an Inverse Warper

by

Lawrence W. Chang

Submitted to the Department of Electrical Engineering and Computer Science on May 22, 1998, in partial fulfillment of the requirements for the degrees of Bachelor of Science in Computer Science and Engineering and Master of Engineering in Electrical Engineering and Computer Science

Abstract

Image based rendering is a new approach to view synthesis significantly different from previous graphics algorithms which use geometric models of a scene. Warping is an image based rendering technique which uses projective geometry and disparity maps to synthesize new views.

Forward warping uses the warping equation to map pixels from a reference image into the desired image. Inverse warping creates a reconstruction of the reference image and then samples the reconstruction to find values for the desired image. With single reference images, forward warping has a significant speed advantage. However, reconstruction and multiple reference image issues are better handled with inverse warping.

The three issues of confidence, ranking, and visibility surrounding the use of multiple reference images in inverse warping are discussed. Five different variations of an inverse warper are presented, each representing a different trade off between speed and correctness.

Thesis Supervisor: Leonard McMillan
Title: Assistant Professor, MIT Computer Graphics Group

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Image based rendering (IBR) represents a new approach to view synthesis. Traditional rendering methods create images from models represented by three-dimensional geometric data. Image based rendering describes a group of techniques that use images as their basic underlying model. Possible advantages of IBR include decoupling image synthesis from the geometric scene complexity, as well as the possibility of rendering new images from real images.

1.1.1 Previous Work

View morphing [6] allows the creation of an image between two other reference images. The technique provides a smooth transition between camera positions using projective geometry. The two reference images are warped before morphing. After the morph occurs, the result is warped again. The morphs occur in parallel views, which maintains consistent shape of the object through the morph. Although constrained in the views it can synthesize, the results are visually realistic.

Techniques like Light Fields [2] and Lumigraph [1] use many images as samples of the light field. These images are constrained in their locations and their fields of view. Although acquiring the images requires specialized equipment, no 3-dimensional data or user specified correspondences are needed. New views are synthesized by resampling the images.

1.1.2 Previous Warping Work

I will refer to image-based rendering techniques that use projective geometry and disparity maps as image warping [4][5]. There are two approaches to image warping. Forward warping takes pixels from the reference image and moves them to the correct place in the desired image. Inverse warping fills in pixels in the desired image by sampling pixels in a reference image. The addition of the disparity map introduces three dimensional information; however, the forward and inverse warping algorithms are not dependent on the three dimensional complexity of the image, which allows for the possibility of rendering techniques which have a complexity dependent purely on the size of the reference image.

1.2 Overview

In this thesis I explore variations on the inverse warping approach to image based rendering. Specifically, I address the problems of incorporating information from several reference images. I will suggest a confidence metric for comparing the quality of the contributions from each reference image, and a heuristic for ordering the set of reference images such that the highest confidence reconstruction will appear in the images considered first by the inverse warping algorithm.

Chapter 2 presents projective geometry, the warping equation, and other fundamental concepts of image warping. The forward and inverse warping algorithms are discussed in Chapter 3. The issues of combining multiple inverse warps will be broken down into three chapters. Chapter 4 proposes a post-warp measure of the actual suitability of the result from an inverse warp for the desired pixel based on comparisons of solid angle. Chapter 5 will discuss the heuristic ranking method, a pre-warp metric to order inverse warps from most to least promising. Chapter 6 describes and explores the visibility problem. I will present results in Chapter 7, and Chapter 8 is composed of future work and conclusions.

Chapter 2

Basic Concepts

2.1 Projective Geometry

Both forward and inverse warping rely on a mathematical relationship expressed in the warping equation. Several basic concepts of projective geometry [5] are needed to understand the warping equation and, in turn, inverse warping.

2.1.1 The Pinhole Model

An image is the end result of a projection of the world (3D objects) onto a plane. Real images are the result of some system of lenses; synthetic images simulate optics in a computer. The ideal pinhole camera represents the standard computer graphics model (See Figure 2.1). Projective geometry provides a convenient way to speak about the mapping of 3D Euclidean points and 2D image space points created by a pinhole camera.

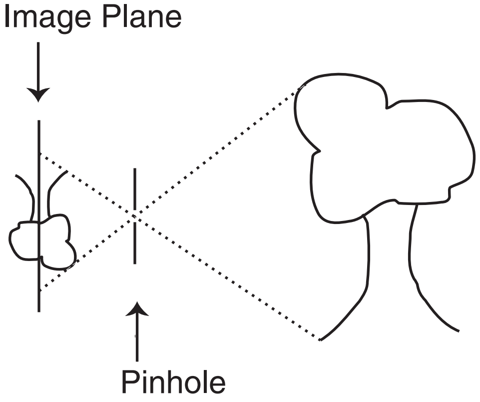

Figure 2.1: Pinhole Camera

In the physical world the image plane is placed behind the pinhole relative to the photographed scene. In computer graphics, the pinhole is also known as the eye point or center of projection. Moving the image plane in front of the center of projection simplifies notation and has no mathematical consequences, despite the incongruity with physical reality.

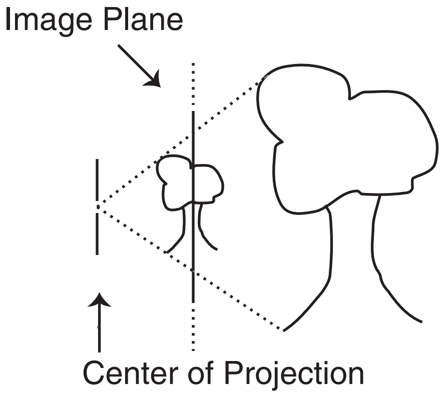

Figure 2.2: Center of Projection and Image Plane

2.1.2 Euclidean to Image Space Mappings

Along any ray from the center of projection into the scene, an infinite number of possible 3D points exist. All points along the ray map to a single point in the 2D image plane. This mapping can be described as a transformation from 3D Euclidean coordinates to 2D projective coordinates. In warping, the transformation of 2D image plane (projective) coordinates to 3D Euclidean is a more fundamental concern (note: projective vectors are always in bold face, all others are 3D Euclidean). The relationship between a Euclidean point on the image plane and its image space coordinates u and v, can be expressed as:

$$\vec{E} = \begin{bmatrix} E_x \\ E_y \\ E_z \end{bmatrix} = \begin{bmatrix} s_x & t_x & w_x \\ s_y & t_y & w_y \\ s_z & t_z & w_z \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = P \cdot \vec{\boldsymbol{q}} \tag{2.1}$$

$\vec{w}$ is the Euclidean vector from the center of projection to the image space origin (typically a corner of the finite image plane). The Euclidean vectors $\vec{s}$ and $\vec{t}$ describe the spacing of pixels in the image plane. The three vectors make up the matrix P.

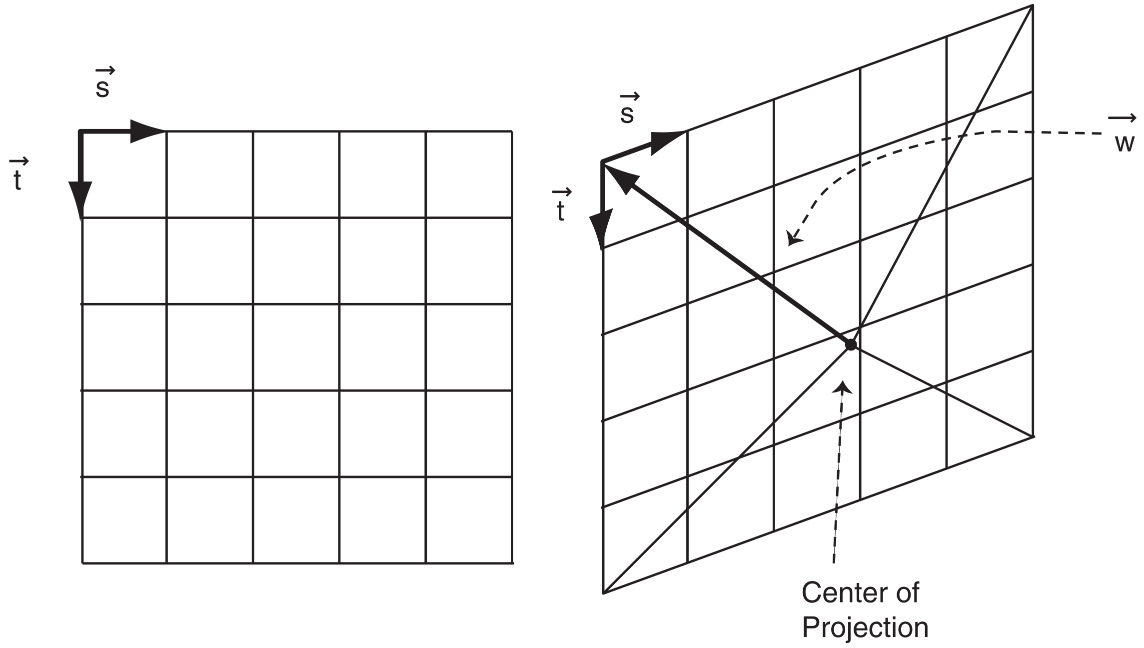

Figure 2.3: Finite Image Plane and Vectors

In order to fully recover a pixel's Euclidean 3D world space coordinates, the world space location of the center of projection is required, as well as some scaling factor d.

$$\vec{Q} = C + d(P \cdot \vec{\boldsymbol{q}}) \tag{2.2}$$

This scaling factor d is a measure of depth relative to the distance from the center of projection to the image plane along that particular ray (i.e. when d is 1, the points lie in the image plane). The parameters of the camera, its view frustum shape attributes, are encoded within P. When this information is combined with the image space coordinates of a specific pixel, we have enough information to constrain the Euclidean 3D coordinates to lie along the ray that begins at the center of projection and passes through the pixel. The depth d at the pixel locates the point along the ray.

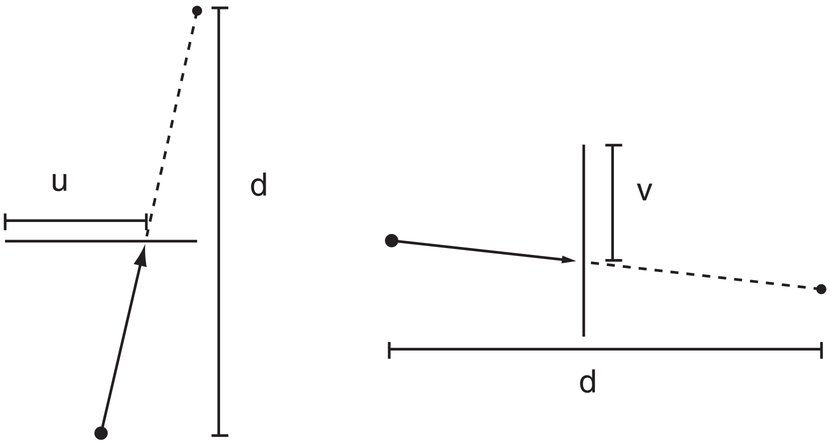

Figure 2.4: Pixel to Euclidean Correspondence

2.2 Warping Equation

Now consider an image to be synthesized (a desired image) and a reference image. For both images, the center of projection and view frustum parameters (i.e. the P matrix) are known. Also, for the reference image, a range value is known for each pixel. Given this framework, finding the desired image coordinates of a desired pixel, $\vec{q_d}$, that correspond to a reference image coordinate $\vec{q_r}$, is a matter of solving this equation:

$$C_R + d_R(P_R \cdot \vec{\boldsymbol{q_R}}) = \vec{Q} = C_D + d_D(P_D \cdot \vec{\boldsymbol{q_D}})$$

$$C_R - C_D + d_R(P_R \cdot \vec{\boldsymbol{q_R}}) = d_D(P_D \cdot \vec{\boldsymbol{q_D}})$$

$$\frac{d_D}{d_R}(P_D \cdot \vec{\boldsymbol{q_D}}) = \frac{1}{d_R}(C_R - C_D) + P_R \cdot \vec{\boldsymbol{q_R}} \tag{2.3}$$

In image synthesis, we are ultimately not concerned with the 3D Euclidean coordinate, but only with the 2D image space coordinates and characteristics of the reference pixel in the desired image. Furthermore, scalar multiples of P produce equivalent image space coordinates. The scaling $\frac{d_D}{d_R}$ term can therefore be ignored. The $\frac{1}{d_R}$ term is called disparity. This disparity is similar to the stereo disparity of computer vision.

$$\overrightarrow{\boldsymbol{q_D}} = \delta_R(u_R, v_R)(P_D^{-1} \cdot (C_R - C_D)) + P_D^{-1} P_R \cdot \overrightarrow{\boldsymbol{q_R}} \tag{2.4}$$

The warping equation (See (2.4)) is, in truth, only equality to a scale factor. Figure 2.5 shows the relationship between the vectors involved.

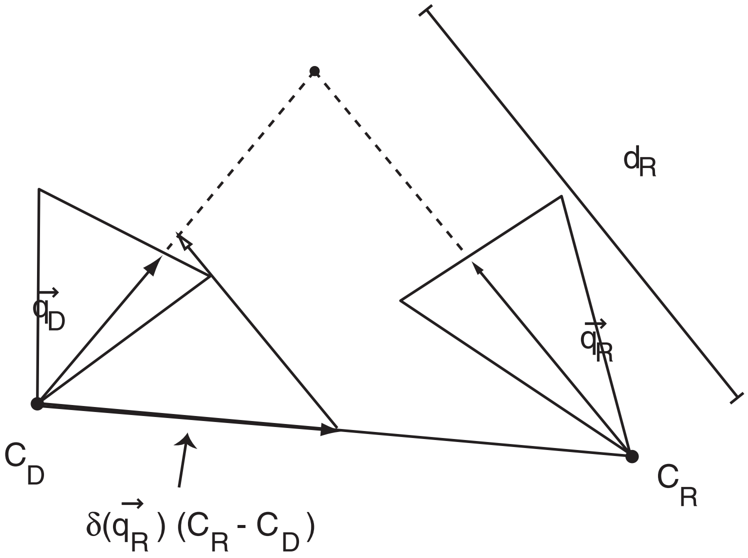

Figure 2.5: The Warping Equation Illustrated

2.3 Epipolar Lines

Figure 2.6 depicts the projection of a ray from one image onto another image's image plane. A ray begins at the center of projection of an image, pierces the associated pixel, and continues to infinity. The projection of the desired image's center of projection on the reference image plane is called the epipole; its image space coordinates are found by solving Equation (2.2) with C_D as $\vec{Q}$

$$P_R \cdot \overrightarrow{q_{epi}} + C_R = C_D$$

$$\overrightarrow{q_{epi}} = P_R^{-1} \cdot (C_D - C_R) \tag{2.5}$$

The infinity point is defined by ray parallel to the desired pixel ray on the reference image plane. The image space coordinates of the infinity point are given by

$$\overrightarrow{q_{inf}} = P_R^{-1} P_D \cdot \overrightarrow{q_D} \tag{2.6}$$

The projected image of the entire ray is a line segment connecting these two points ($\overrightarrow{q_{epi}}$ and $\overrightarrow{q_{inf}}$)and is known as the epipolar line. This line represents the loci of points that could intersect with the pixel ray, as seen from the reference image.

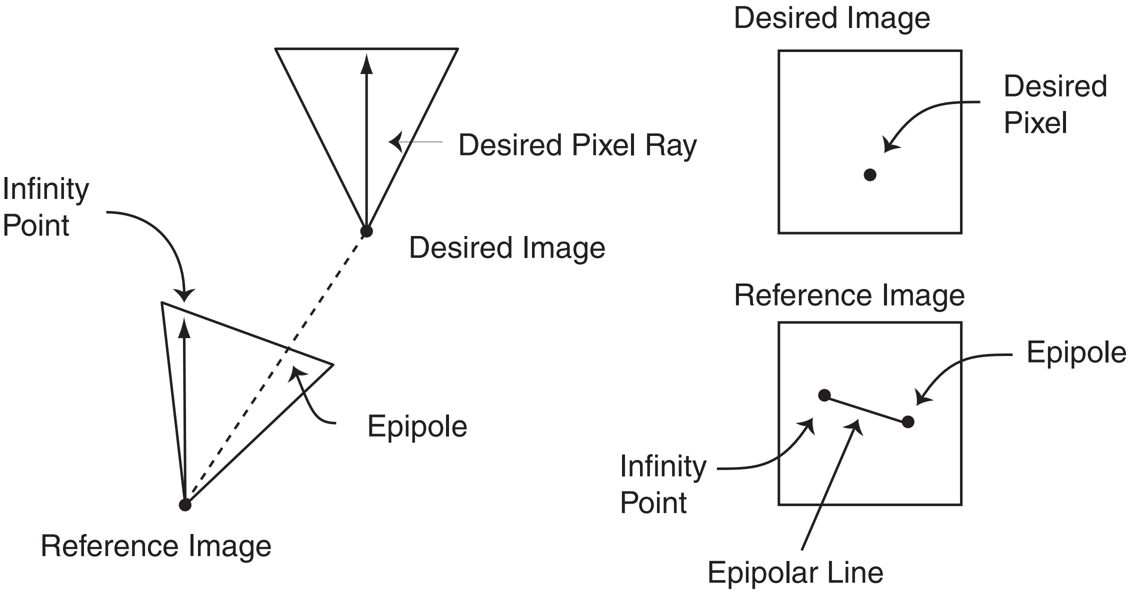

Figure 2.6: An Epipolar Line

In this simple case, if we wish to traverse the line from front to back, we begin at the epipole and move toward the infinity point. When the epipole or infinity point corresponds to a point that lies behind the center of projection of the reference image, they are termed antipodes. When the epipole is an antipode, but the infinity point lies in front, the epipole still defines the epipolar line, but instead we begin traversal at the edge of the image opposite the infinity point and move towards the infinity point.

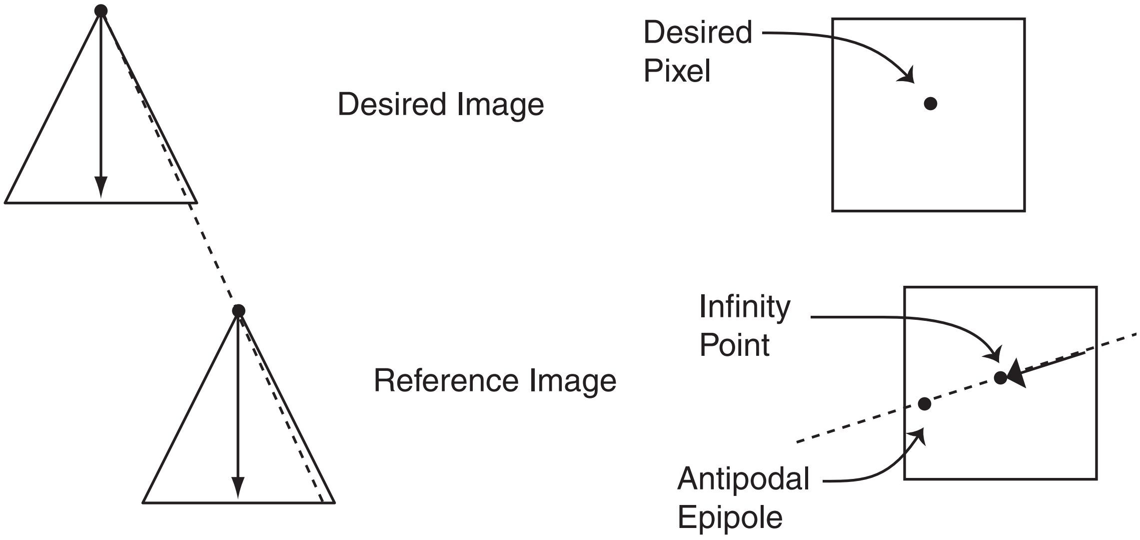

Figure 2.7: Antipode Epipole

If, on the other hand, the infinity point is an antipode, we begin at the epipole and move on the line away from the infinity point.

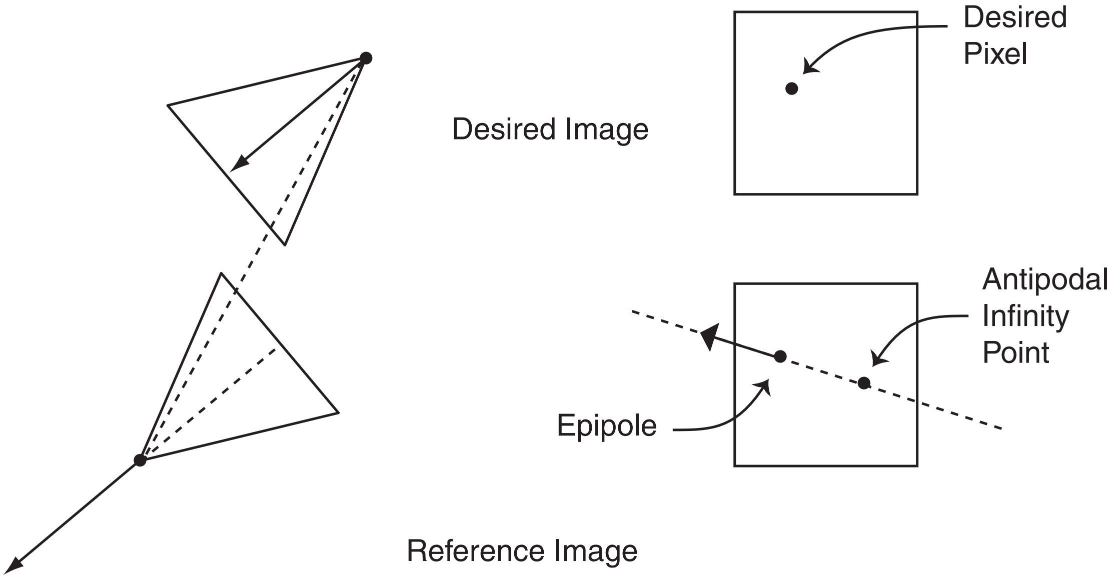

Figure 2.8: Antipode Infinity Point

If both the infinity point and epipole are antipodes, no pixels of the reference image could possibly correspond with that desired pixel (the ray is always behind the reference image).

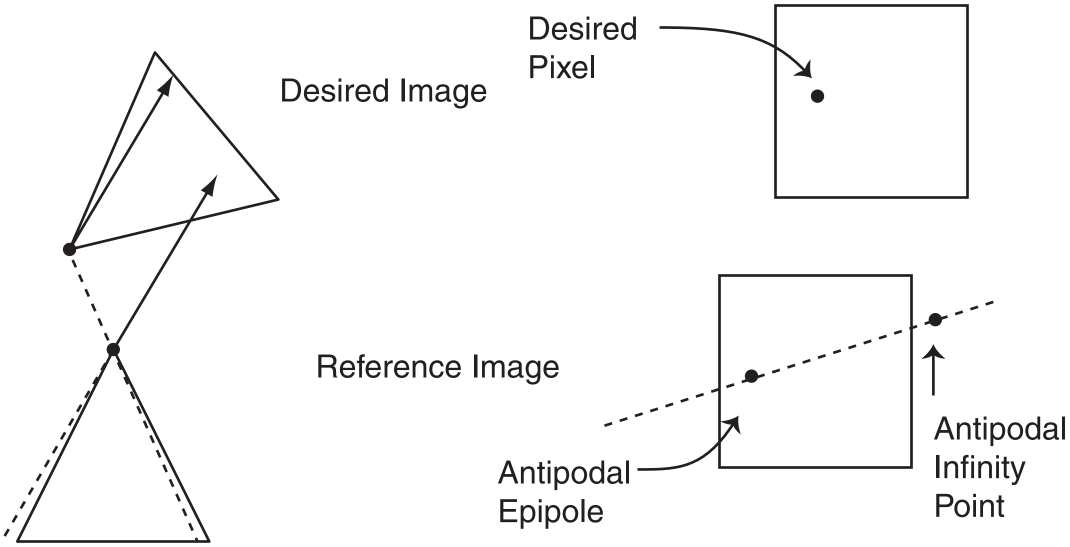

Figure 2.9: Antipode Epipole and Antipode Infinity Point

Chapter 3

Warping

3.1 Forward Warping

Applying the warping equation to the image space coordinates of the pixels of the reference image produces image space coordinates of the desired image. Applying the warping equation is constant work for a pixel, so the complexity of the algorithm is O(n x m), where n and m are the width and height of the reference image.

Matters, however, are not quite so simple. An image is a discrete sample of a continuous function[1], where each pixel represents the light collected over some solid angle. This solid angle is represented by the area of the pixel in image space. The solid angle that a reference pixel represents is different in the desired image. The solid angle (and in turn the area or size that the pixel should fill) could increase or decrease in the desired image. If it becomes smaller, it may not fill the entire area of a pixel in the desired image and a desired pixel should be some sort of combination of several reference pixels. If it becomes larger, it will spread over several pixels, some of which are only partially filled.

Reference pixels could map onto the same pixels. When this occurs, care must be taken that the first visible pixel along the ray is chosen. If the pixels are warped in a back-to-front ordering, then successive pixels may simply overwrite previously written pixels. If the epipole of the desired image on the reference image is not an antipode, the pixels should be traversed toward the epipole. If the epipole is an antipode, forward warps should proceed from the epipole outward.

Forward warping makes the reconciliation of multiple reference images difficult. If two reference images have pixels that map to the same desired image, the depth values are

---

[1] At least, any synthetic or digitized image.

required to correctly choose which pixel is visible. Most importantly, it is difficult to know which portions of a desired image a reference image will fill without actually warping the entire image. Extracting all relevant pixels from a set of reference images requires the forward warp of all pixels in all reference images.

The algorithm for single image forward warping, when reconstructing naively (each reference pixel corresponds to only one desired pixel), is straight forward. The pseudocode for single reference forward warping follows:

```
begin
epipole = COP of desired image on reference
using epipole, order pixels in reference from back to front
foreach pixel in reference, using ordering
    u,v = warp reference pixel
    if (u,v is not clipped in desired)
        desired pixel at u,v = reference pixel
end
```

3.2 Inverse Warping

The reference pixels that correspond to a desired pixel are constrained to lie along the epipolar line defined by the projection of the desired pixel's ray onto a reference image. By traversing this line from front to back (See 2.3 Epipolar Lines) the first intersection found is the first surface intersection[1]. The disparity necessary for a reference pixel to match is linear rational along the epipolar line. Parameterizing the line to calculate the required disparity is simple. How is the final reconstruction accomplished? (i.e. how is a color value for the desired pixel computed after a disparity match or intersection found?) In this work,

---

[1] The first intersection is only first intersection the reference image contains. Hidden occluders can cause visibility problems that require multiple reference images to resolve correctly. See Chapter 6 for more on visibility concerns.

a group of four reference pixels are used to define the four corners of a bilinear surface. Final reconstruction finds the point of intersection on this surface, and interpolates to find the color value.

The inverse approach has two inherent advantages. It allows work to be focused narrowly; an inverse warp returns possible pixel matches for a specific desired pixel, traversing only pixels along the epipolar line. Secondly, more information is available for reconstruction. This information can be used to help evaluate the appropriateness of a reference pixel (or the set of four reference pixels defining the bilinear surface) for each desired pixel. When dealing with multiple reference images, work for a desired pixel can be bounded, and more information is available for combining inverse warps from several reference images. The following is pseudo code for a single reference inverse warp:

```
begin
epi = desired COP projection on reference
foreach desired pixel p(u,v)
    inf = p(u,v) on reference
        while no intersection & not end of epi line
            traverse epipolar line: epi_i to inf_i
    p(u,v) = intersection
end
```

In the single reference image case, if many of the pixels of the reference image are relevant to the desired image (i.e. the epipolar line is long), inverse warping can incur a performance penalty in the extra work required to traverse the epipolar lines.

3.3 Framework for Multiple Reference Images

The algorithm for handling multiple reference images adds a loop to the single inverse warping algorithm. The framework for multiple reference inverse warping:

```
begin
foreach reference image i
    epi[i]= desired COP projection on i
foreach desired pixel p(u,v)
    foreach reference image i
        inf[i] = p(u,v) on i
        while no intersection[i] and not end epiline
            traverse epipolar line: epi[i] to inf[i]
    if (at least one intersection[i] found)
        p(u,v) = best of all intersection[i]
end
```

This thesis evaluates five variations on the inner while loop of this algorithm. The concept of multiple intersections from a single inverse warp will be introduced. A heuristic method will be described that will exit early out of the while loop if the confidence of a returned pixel is above a threshold. Finally, several different criteria for determining the "best" intersection of all results are discussed and implemented.

Chapter 4

Post-Warp Confidence

4.1 Assumptions

Confidence is a post warp measure of a reference pixel's suitability for the desired pixel it has warped to. To understand the confidence metric, an examination of the underlying model of image points in the reference image is necessary.

Each pixel value of a reference image represents the integration of incoming light for the solid angle which the pixel subtends. If the reference image is the result of a raytracer which super samples each pixel, the pixel may represent several samples within the solid angle. If the reference image is a real image (i.e. the result of a physical camera), the pixel represents the integration of all the photons that were collected by the sensor. The reconstruction of the reference image using bilinear surfaces allows new pixel samples to be taken. Reconstruction in this manner appears as a "skin" formed by the bilinear patches. When the depth variations of neighboring reference pixels is very large, the skin created can project to a large area when viewed from other viewpoints. This area is maximum for sight lines that are nearly normal to the bilinear surface. Figure 4.1 and 4.2 show an example original and reconstructed surface.

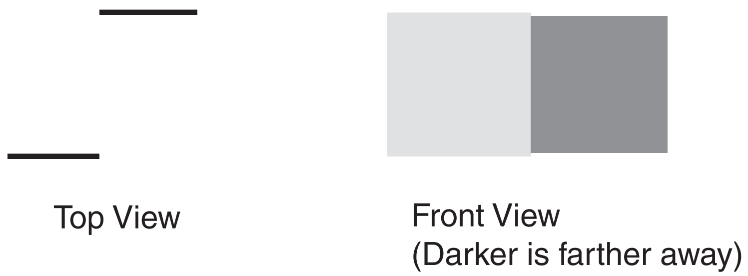

Figure 4.1: Real Surface

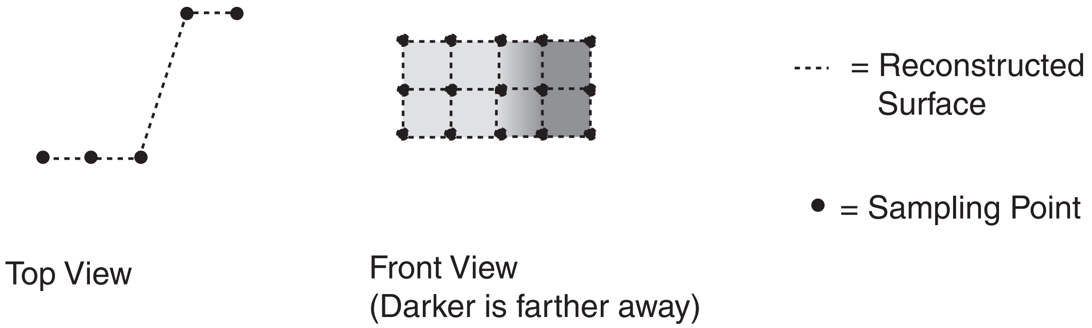

Figure 4.2: Desired View and Reconstructed Surface

Consider warping from a reference image to a desired image in the same location and with the same look direction. The reconstructed surfaces subtend the same solid angle, and the reconstructed surfaces are perfect for the image.

The general outlines of the confidence metric are then clear. Confidence should be high when the solid angle represented by the reconstructed bilinear surface is close to the solid angle of the desired image.

4.2 Solid Angle Computations

The magnitude of a solid angle Γ, in steradians, of an object viewed from a point is the ratio of a fraction of the surface area of the sphere to the sphere's radius squared. The fraction of the surface area of the sphere is calculated by finding the ratio of the area of the radial projection of that region onto the sphere to the surface area of the sphere (See Equation (4.1)).

$$\Gamma = \frac{S}{r^2} \tag{4.1}$$

The question now becomes finding the radial projection of a region and the surface area of that projection. Both the desired pixel and the boundaries of the bilinear surface can be described in the image space coordinates of the desired image. Returning to Equation 2.1, the radial projection of an image space point on the unit sphere whose coordinate system is centered at the image's center of projection is the Euclidean coordinates of that point normalized.

$$P \cdot \vec{q} = \vec{Q}$$

$$\frac{\vec{Q}}{|\vec{Q}|} = \hat{Q} \tag{4.2}$$

Consider the case of a pixel near the center of the image. The stretching effects of the pixel's radial projection are minimized. In this case, the four corners of a pixel, $\hat{Q}_0$, $\hat{Q}_1$, $\hat{Q}_2$, and $\hat{Q}_3$ create a patch upon the sphere's surface. If the h in Figure 4.3 is small, the surface area of such a patch may be approximated by the area of the surface created by the bilinear surface of these points.

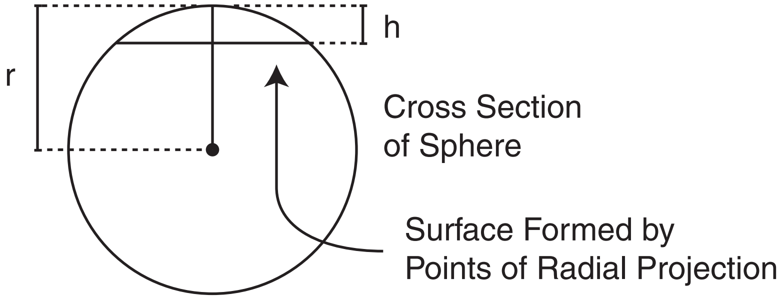

Figure 4.3: Surface Formed by Radial Projection

The ratio of the area of the radial projection to the surface area of the unit sphere can be approximated as:

$$\frac{|(\hat{Q}_0 - \hat{Q}_1) \times (\hat{Q}_1 - \hat{Q}_3)|}{4\pi} = S \tag{4.3}$$

Since the unit sphere has a radius of 1, *S* is the approximation of the magnitude of the solid angle. The ratio of the magnitude of two such patches is:

$$\frac{S_{pixel}}{S_{bilerpsurface}} = \frac{\dfrac{\left|(\hat{Q}_0 - \hat{Q}_1) \times (\hat{Q}_1 - \hat{Q}_3)\right|}{4\pi}}{\dfrac{\left|(\hat{Q}_{0bilerp} - \hat{Q}_{1bilerp}) \times (\hat{Q}_{1bilerp} - \hat{Q}_{3bilerp})\right|}{4\pi}} = \frac{Area_{pixel}}{Area_{bilerp}} \tag{4.4}$$

4.3 Solid Angle and Shape

Unfortunately, the use of only the magnitude of the solid angle as a confidence metric has a significant drawback. It is possible for two objects to have the same scalar solid angle magnitude, but have very different radial projections.

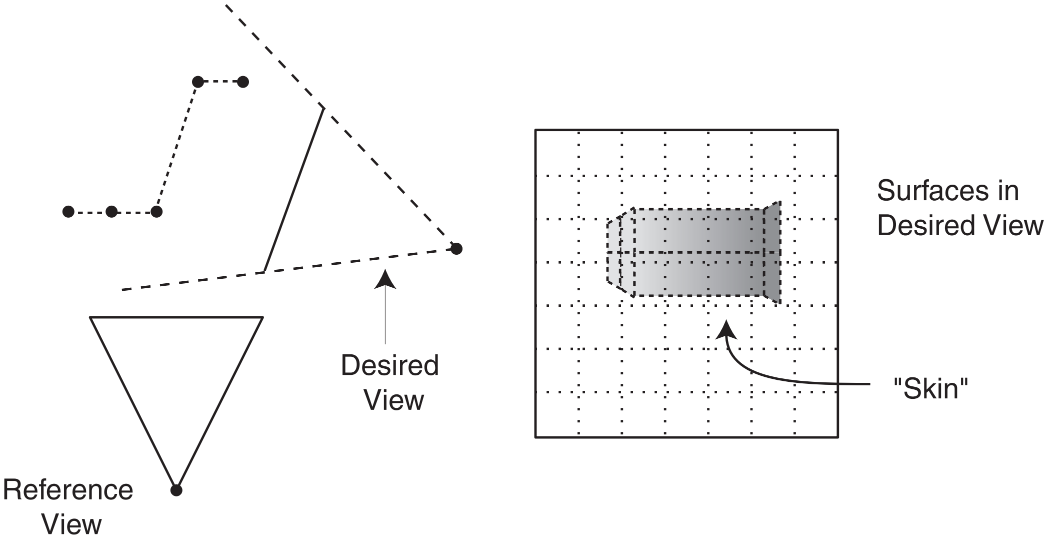

Figure 4.4: Confidence and Shape

Returning to the bilinear surface with a significant difference between the depth of 2 opposite edges, the compression of the height and the stretching of the width of the projected bilinear surface could produce a scalar solid angle very close to the solid angle of the desired pixel. But the obvious stretching of the pixel makes it a bad choice for the desired pixel. The confidence metric must therefore also take into account the difference in shape between the desired pixel and the bilinear surface.

4.4 Implementation of Confidence

The previous sections assumed that the desired pixel and projection of the bilinear surface into the desired image were both close to the center of the image. Solid angle computations away from the center of the image must deal with greater stretching by the radial projection onto the sphere, making calculations more complicated. Already, four forward warps were necessary to find the desired image space coordinates of the bilinear surface. The confidence metric is becoming more and more costly to compute.

4.4.1 Comparative Area and Squareness

In the interests of efficiency, several simplifying steps were taken. If we assume that all pixels subtend equivalent solid angles, which is approximately true for an image with a very narrow field of view, comparison of solid angle magnitude can be simplified to comparisons of the projected area of the bilinear surface to the area of the reference pixel. Because this comparison is done in the image space of the desired image, the area of a pixel is 1[1]. When the area of the projected bilinear surface is larger than the pixel, the comparative area is $\frac{1}{Area_{surface}}$. When smaller, it is $Area_{surface}$. This formulation limits the solid angle magnitude measure from 0 to 1, 1 as a perfect match. The shape component of confidence is an attempt to measure how similar the shape of the bilinear surface is to the original square pixel[2]. Holding perimeter constant, quadrilaterals have a maximum area when they are squares. Therefore, a measure of a quadrilateral's closeness to a square is $\frac{4\sqrt{Area}}{Perimeter}$. This metric is again limited to the range from 0 to 1.

4.4.2 Combining Area and Shape into Confidence

The remaining issue is the combination of the similarity of solid angle magnitude and shape, formulated through comparisons of area and shape, into a final confidence measure.

---

[1] Since the u and v length of a pixel in image space is, by definition, 1, the area of a pixel is also 1.

[2] Again, because these comparisons are made in image space, each side of a pixel is 1, therefore in image space, no matter what the parameters of the camera, a pixel is square.

The confidence metric used in the algorithms presented later is the product of comparative area and squareness.

Chapter 5

Pre-Warp Ranking

5.1 Goals of Ranking

As a pre-warp operation, the goal of ranking is to find the reference image most promising for each desired pixel. The most promising image is the reference image that has the most similar ray to the desired pixel.

5.2 Euclidean Intuition

Approaching the ranking problem from a "Euclidean" perspective, the two intuitive measures of similarity are the dot product of the ray representing the desired pixel with the look vector[1] of a reference image and the distance between the centers of projection of the desired and reference images.

These two metrics can be misleading. The reference image R1 in Figure 5.1 scores well on both metrics when compared with the desired pixel illustrated. The image R2 is actually more likely to contain a match for the desired pixel, despite its lower distance score. A good combination of the two metrics could entail essentially clipping the desired

---

[1] The look vector of an image is defined here as the vector from the center of projection to the image plane and is perpendicular to the image plane.

ray against the reference frustum before ranking. The best method for combining these two measures into one unifying ranking score for a reference image is unclear.

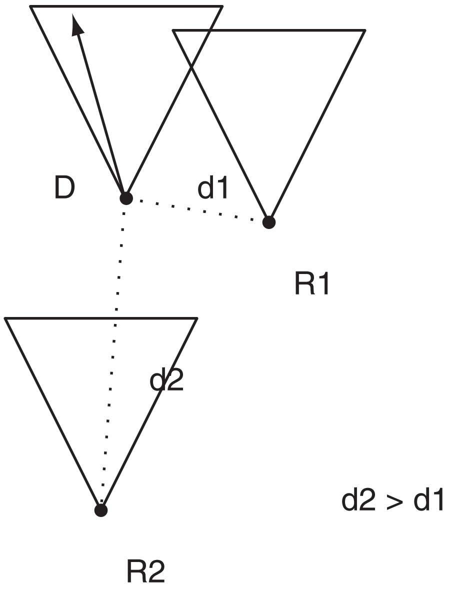

Figure 5.1: Misleading Euclidean Intuition

5.3 Epipolar Line Length

Moving back into the framework of projective geometry, the length of the epipolar line corresponding to a desired pixel on the reference image provides many of the same qualities as the measures discussed in the previous section. Consider the epipolar line lengths of the desired pixel on the two references of Figure 5.2

Reference R2 has a viewing direction that is almost perpendicular to the desired pixel ray; its epipolar line is correspondingly long. R1 has a better viewing direction, and the length of the epipolar line is shorter.

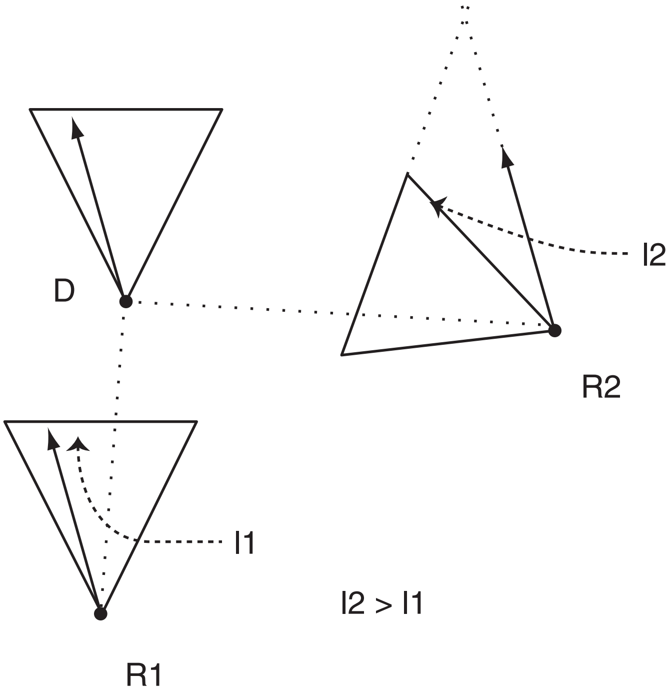

Figure 5.2: Simple Epipolar Line Length Example

5.3.1 Disparity Clipping

Unfortunately, the epipolar line length heuristic fails in several cases. For example, when a reference image R2 lies on the line formed by the centers of projection of R1 and the desired image, and R1 and R2 share the same look direction, then the epipolar line length is the same. However, R1, the image closer to the desired image, should be ranked higher.

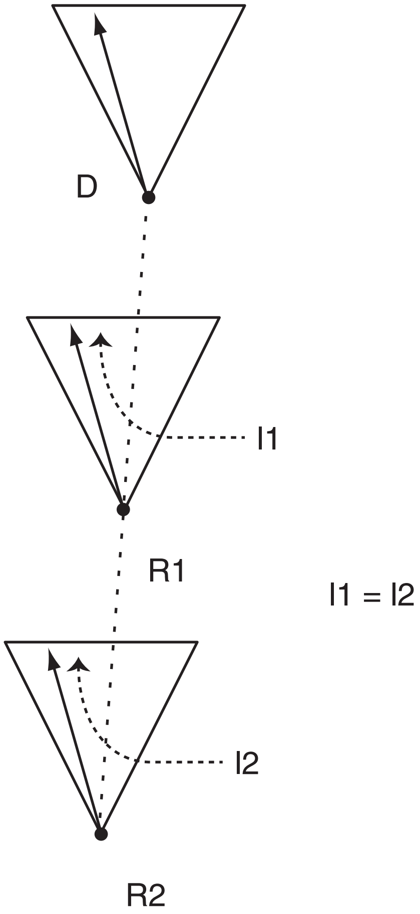

Figure 5.3: Near/Far Reference Images

Similarly, when two images share a similar displacement, but radically different look directions, their epipolar line lengths can both be infinitely long, although one image is clearly more promising than the other. Consider R1, which has the look direction of the desired image, but has a center of projection transposed away from the desired image. R2 has the same center of projection, but a look direction perpendicular to R1 and the desired image. The epipole of the desired image upon R1 is at infinity on the image plane. The length of the epipolar line in this case is infinite. For R2, a desired pixel ray which is exactly parallel to the image plane will also have an infinite epipolar length[1].

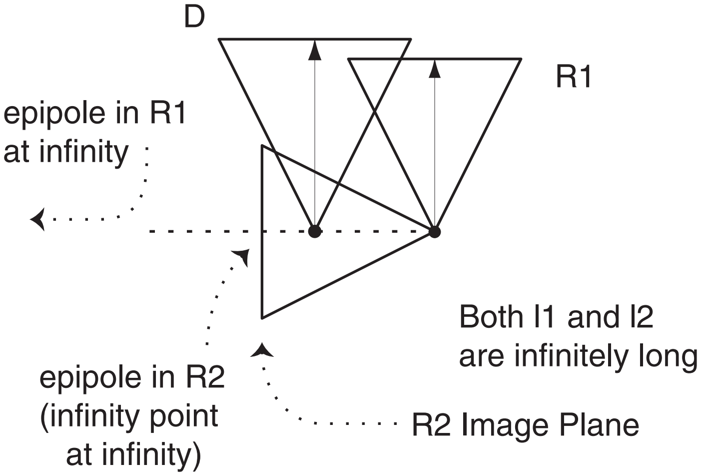

Figure 5.4: Slightly Displaced Reference Image

A slightly altered epipolar line length metric was used in order to overcome these difficulties. Instead of using the full extent of the epipolar line as a metric, the line is first clipped against the "near" and "far" extent of the image. The near extent is defined by the pixel with the least depth and the far by the pixel of the greatest depth. Taken together, the far and near extent define the infinite slab in which all of the pixels, and hence all the surfaces, of the image lie. Because inverse warping requires the calculation of a parameterization of the epipolar line and the corresponding disparity change along that line, clipping using the disparity value is simple. These calculations can be handled as a preprocessing stage, and the results of these calculations can be stored for use when the inverse warp is performed.

---

[1] In the case where the pixel ray direction is not parallel to R2's image plane, a long, but finite epipolar line will result. For this pixel, R2 will rank higher than R1, which is not desirable.

Revisiting the example reference/desired image setups, the simple setup achieves the correct ordering.

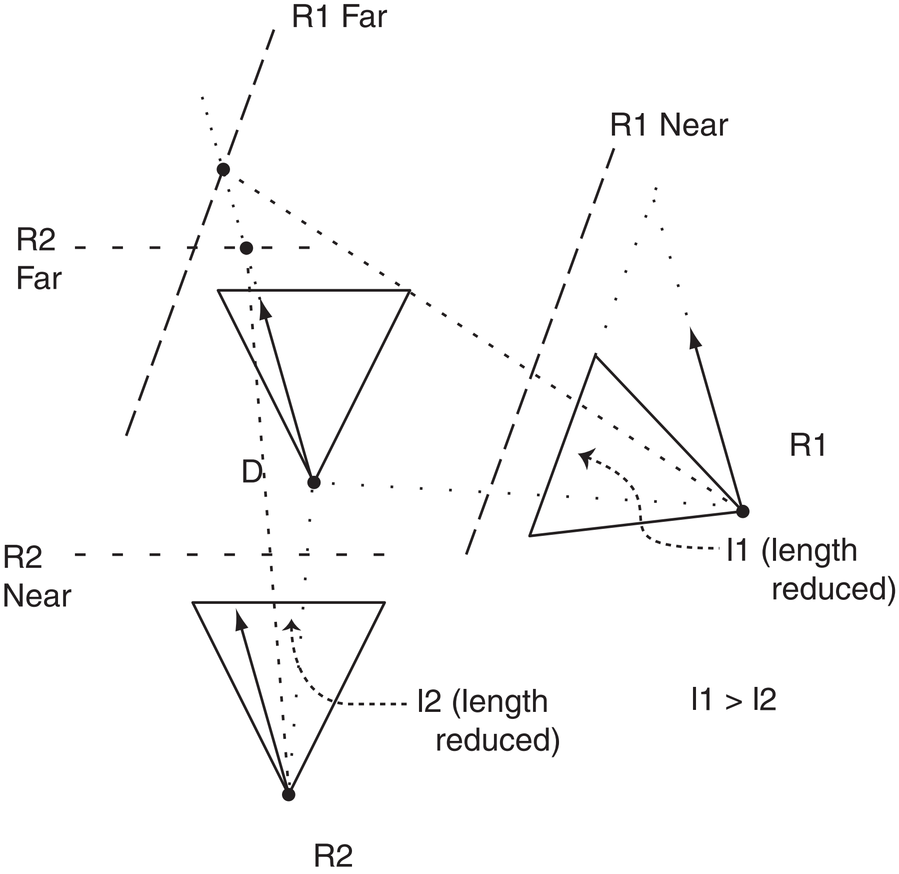

Figure 5.5: Simple Epipolar Line Length Example Revisited

The ranking of the slightly displaced setup is also corrected to give a better ordering.

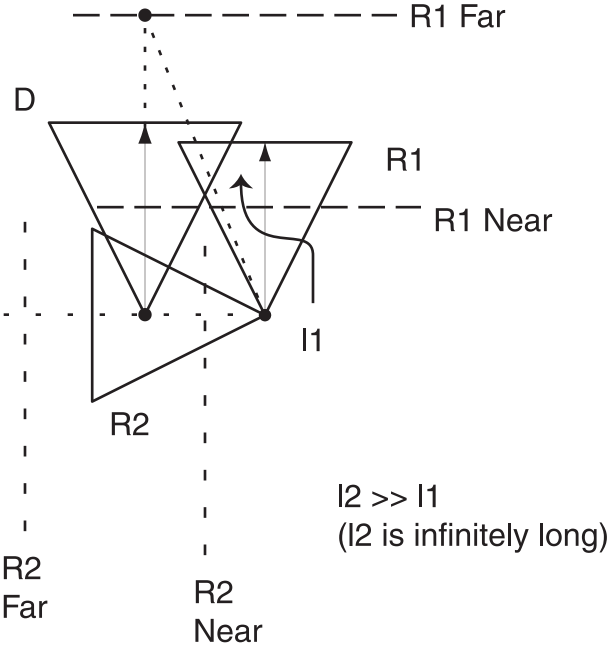

Figure 5.6: Corrected Line Lengths in Slight Displacement Case

In the setup with the center of projections of the reference images in a line, disparity clipped line lengths could produce some erroneous results. When the slab representing the surfaces of R2 does not contain any part of a desired ray, the disparity clipped measure detects the impossibility of an intersection and avoids unnecessary work. If there is overlap, or in the worst case only a very slight overlap of the clipped desired ray within the reference image, the clipped line length could actually be shortened significantly, increasing R2's ranking rather than downgrading it. In this case, this metric will rank these images higher than they should be; however, because the line is severely cut off by near and far clipping, the inverse warp traverses a short line segment, reducing the amount of work "wasted".

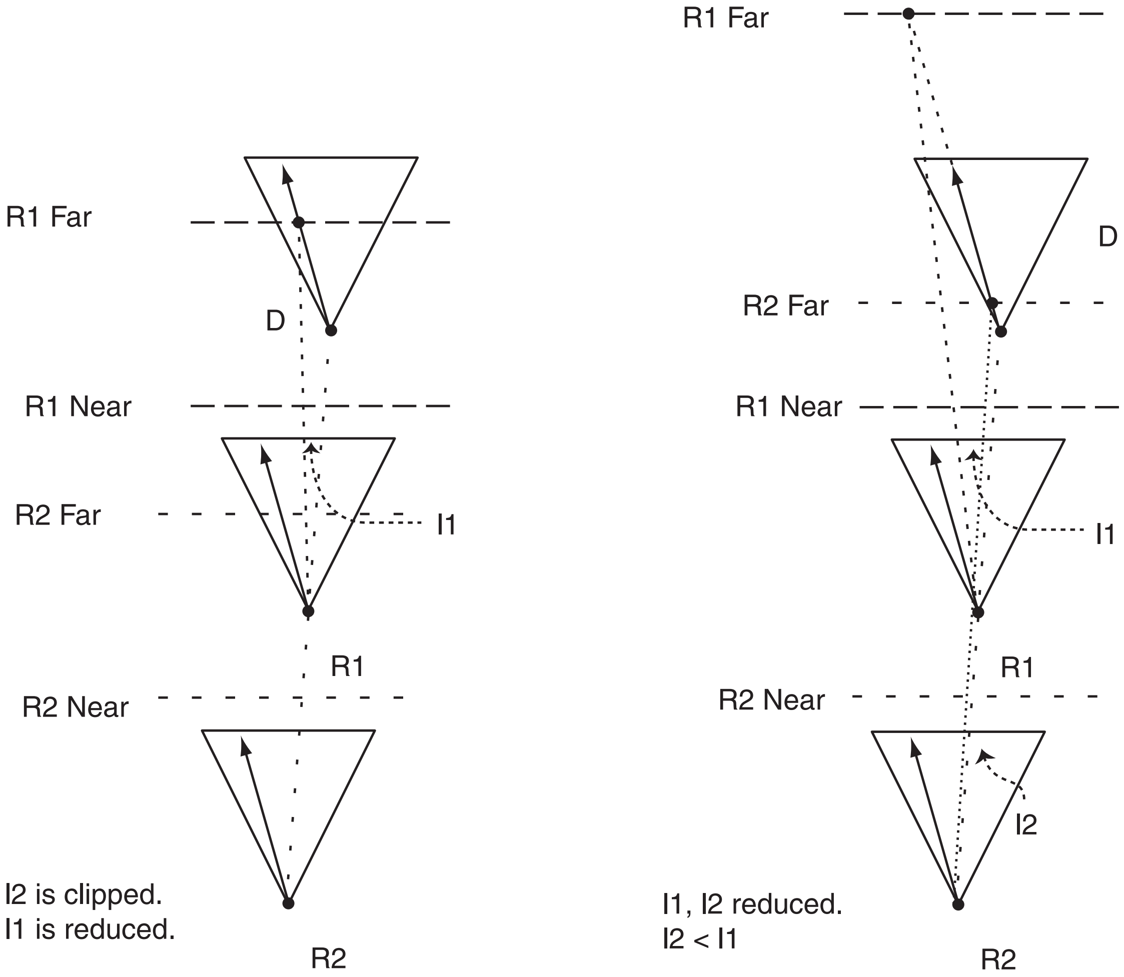

Figure 5.7: Possible Erroneous Cases

Perhaps the most important stumbling block of the disparity clipped epipolar line metric is its bias toward reference images that are "flat." In figure 5.8, R1 and R2 are displaced the same distance from a desired image; the difference in direction and scene composition could result an R1's slab which is much thinner than R2's.

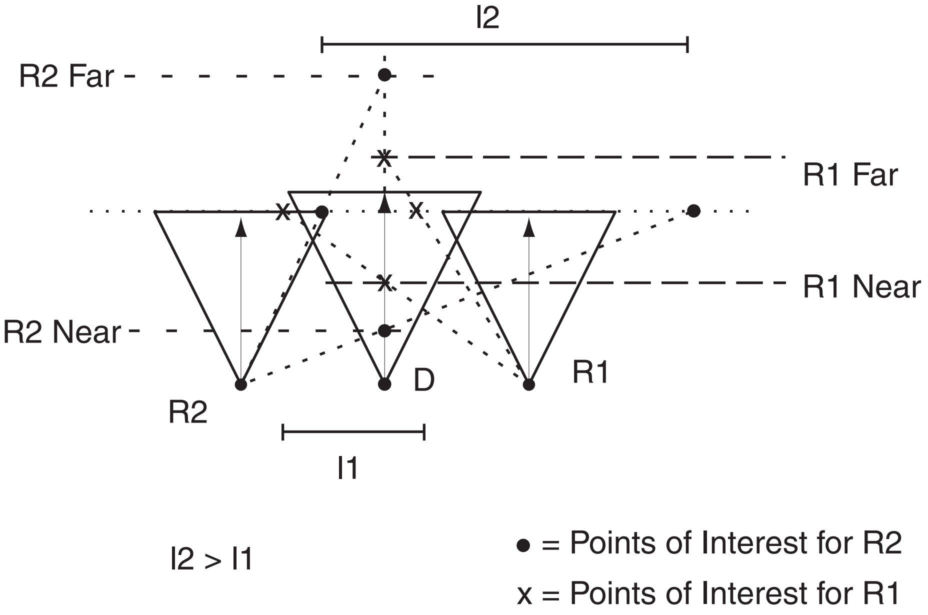

Figure 5.8: Disparity Bias

This bias is undesirable, because R2 is in fact more promising than R1, but the differences in the slab makes the line length of R1 shorter. Therefore, rather than clip against the specific slab of each image, clipping for ranking is done against the slab formed by the nearest and farthest disparity of all reference images.

5.3.2 In Practice

This discussion has ignored the ranking of situations where the dot product of the reference image look direction with the desired pixel ray are negative. In these situations, the reference image is looking in the opposite direction as the pixel ray. In such cases, the epipolar line length ranking is backwards; longer lines should be ranked higher, and shorter lines ranked lower. These cases can be handled by making note of the dot product of the viewing direction of the reference images during ranking, and inverting the ordering. Positive dot product references are always better than negative ones, another aspect of globally disparity clipped epipolar line length in usage.

In the remainder of this work, disparity clipped epipolar line length or epipolar line length ranking will refer to the global disparity clipped epipolar line length ranking.

Chapter 6

Visibility

6.1 Single Inverse Warp Visibility

Visibility issues in inverse warping from a single reference image are relatively straightforward. Traversing the epipolar line from front to back, usually from epipole to infinity point if neither is an antipode, guarantees that earlier matches on this line are closer to the center of projection of the desired image. When reconstruction is performed using bilinear surfaces, this first match may correspond to a real surface in the scene, or it may be an artifact of the reconstruction process between two unconnected surfaces. We call this artifact a *skin*. If the line traversal is not terminated at the first intersection, several matching surfaces can be found.

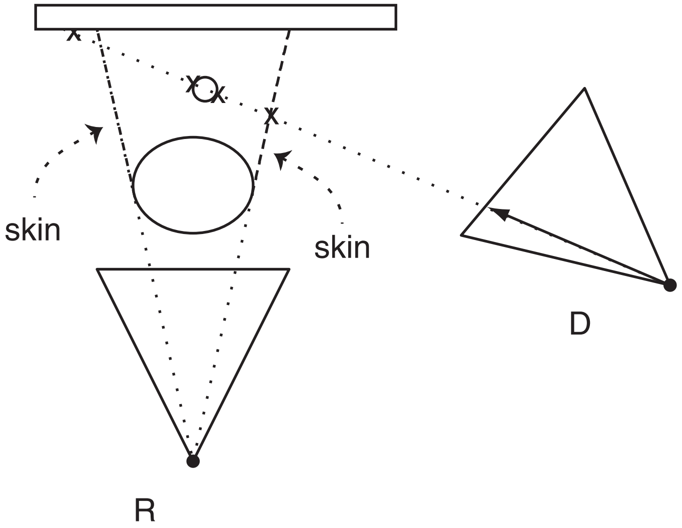

Figure 6.1: Multiple Intersections

Each "in" intersection has a corresponding "out" intersection. The first intersection corresponds to the ray entering an object, possibly created by the skin, and the out intersection corresponds to the ray leaving the "volume" of that object. If a real object, not an object created by the artifacts of reconstruction, lies within this “volume,” it is not visible in the reference image; it is obstructed by other objects in the reference image.

Therefore, if an inverse warper is augmented to return multiple intersections, the first match of a pair is a possible valid surface. For the interval between an in and an out, the possibility of unseen objects exists (See Figure 6.1). Taking the first intersection assures that the first possible visible surface (in the desired image) of the surfaces represented by the pixels of the reference image is used. However, this surface may be a skin.

6.2 Hidden Occluder

A hidden occluder is an object that lies outside the frustum of a reference image, but should be visible in the desired image. Hidden occluders are unavoidable with finite planar reference images, if their look directions and centers of projection are unconstrained.

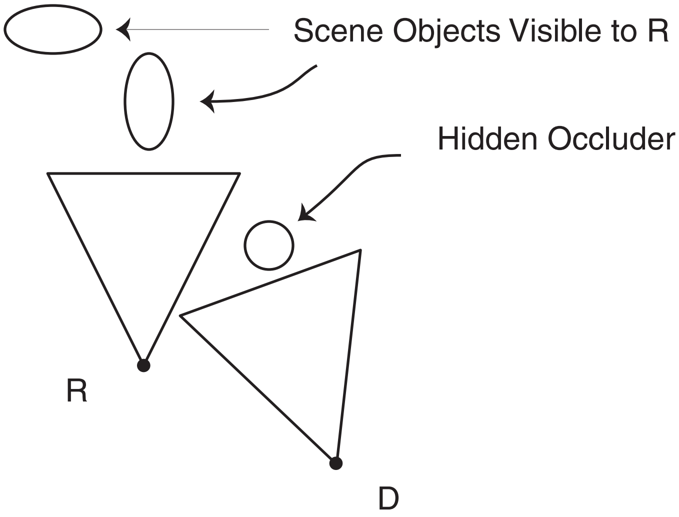

Figure 6.2: Hidden Occluder

6.3 Ramifications

There are two considerations of visibility that must be dealt with in combining multiple reference images. A way must be found to meaningfully compare the results of one reference image with another. Secondly, multiple intersections may be used to disregard skins.

6.3.1 Visibility Line Segment

One way to compare results from multiple reference images is to calculate the depth as well as image coordinates of the new pixel in the desired image. This has the unfortunate affect of adding computation to the warp algorithm. More importantly, calculating the Euclidean coordinates defeats the warping paradigm of performing calculations in image space.

Instead of computing Euclidean depth, visibility may be resolved in the projective framework by forward warping the resulting pixels into one of the reference images. These forward warps produce pixels that lie on the epipolar line of the desired pixel in that reference image[1]. Assuming that this epipolar line is parameterized in t, 0 at the epipole and 1 at the infinity point, lower t values are closer to the desired center of projection. Meaningful comparisons of the first intersection from different reference images are now easily accomplished.

Each in-out pair returned by multiple intersection inverse warping corresponds to an interval on this line. Outside of the interval, no intersections may occur. These blank intervals can invalidate intersections from other reference images; these invalidated intersections must be skins, as shown in Figure 6.3 R1 has three intersections, two ins and one out illustrated. R2 is able to invalidate the first interval represented by the first pair of intersec-

---

[1] Note that when the reference center of projection lies on the line formed by the desired image's center of projection and desired pixel ray, the epipolar line is not a line, but a point. This visibility resolution scheme fails in this case, but visibility calculations can be done in a different reference image.

tions of R1. R2 does not have a final intersection for the desired pixel, but now the second intersection of R1 can be used.

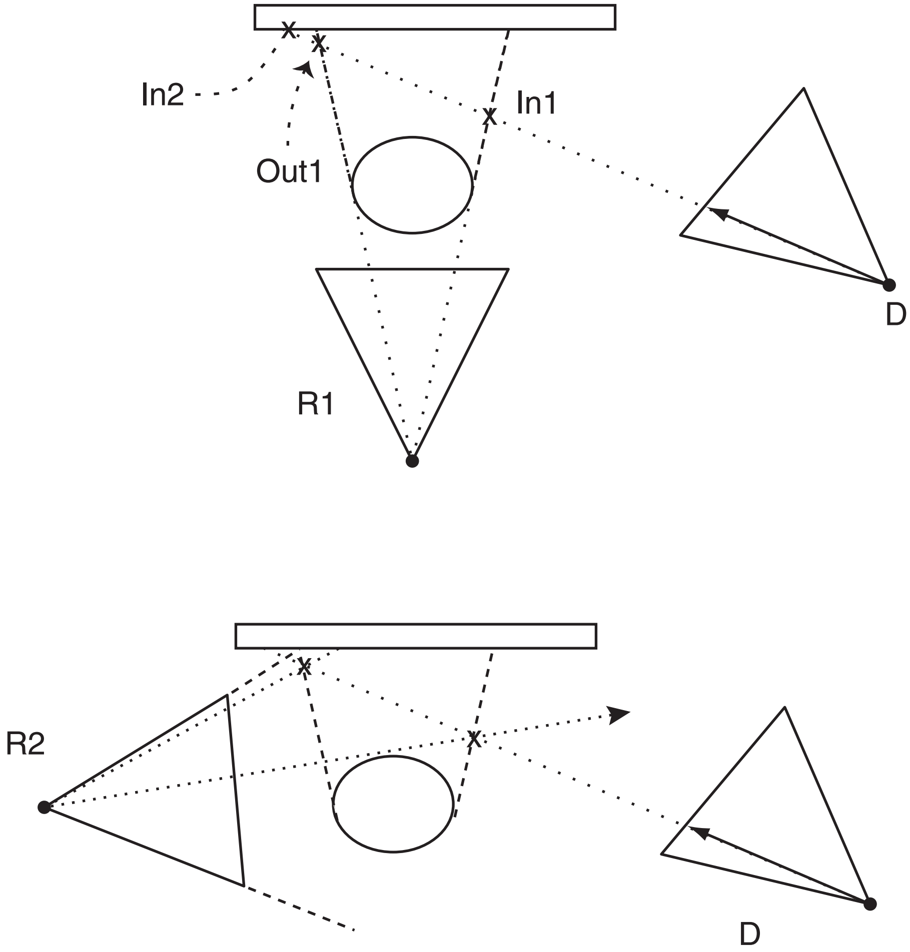

Figure 6.3: Multiple Intersections and the Visibility Line

The visibility line allows R2 to invalidate the first intersection of R1, a skin, and use the third intersection (the second "in" intersection). Note that R2's view of the correct surface would have a low confidence and would not be as accurate as the intersection in R1.

6.3.2 Visibility and Epipolar Line Length

The disparity clipped epipolar line length ranking does not account for visibility. It is possible for a highly ranked reference image to return a high confidence answer that does not correspond to the first visible surfaces represented in the entire set of reference image. With finite planar reference images, it is not possible to detect this situation without warping all reference images. For a discussion on how this problem may be resolved, see the section on visibility in Chapter 8.

6.3.3 Complexity

The complexity of an inverse warp for a single pixel is O(k), if the reference images are square images with k by k pixels. The setup of an inverse warp, finding the epipolar line and its parametrization, is constant for each pixel. The number of pixels traversed is at most the diagonal length of the reference, O(k). Finding multiple intersections does not change complexity, since multiple intersections simply force all O(k) pixels to be traversed on the epipolar line, unlike single intersection warping which allows traversal of the line to terminate at the first intersection. Inverse warping an entire desired image is then O(k*m), where m is the number of pixels in the desired image.

Forward warping the results of multiple inverse warps into a reference image is constant for each match. Enough information is now available for single intersection resolution. Note, that scene complexity has not entered into complexity considerations. Inverse warping without multiple intersections is not dependent on scene complexity. However, resolving the intervals produced from multiple intersections from multiple reference images introduces an element of scene complexity into inverse warping, an unfortunate side effect of the visibility-line technique.

Chapter 7

Results

7.1 Implementation

The base inverse warping algorithm was implemented with another graduate student, Robert Marcato. Initially, a straightforward implementation of the inverse warper was created, with no speed optimizations in the warper itself. The warper was written in C++ to run on SGI workstations. The user interface was created with RapidApp, an SGI tool for creating graphical user interfaces. Inventor was used to create a 3D representation of the reference and desired image frustums and locations within the scene. A snapshot of the scene view is included with each example. Warping code was separated from the user-interface code as much as possible.

Rayshade, a widely available public domain raytracer, was used to create the reference images. Rayshade was modified to provide the depth values and camera parameters needed for warping. A utility program converts these depth values into disparity. These disparity maps are quantized to 255 values. A separate “reference” file lists the names of the image input files, the disparity map files, camera parameters, as well as the minimum and maximum disparity values used in the quantization.

I used this basic inverse warper to explore the heuristic and confidence metrics and create new methods for combining the results of inverse warps. Five algorithms will be explored in the example scenes that follow.

The first method is a naive method. It ranks the reference images for each pixel, and returns only the first result received from the inverse warps. The pseudo code for this algorithm was given in Chapter 3.

I call any method that, for each pixel, continues warping until a confidence threshold is met a heuristic method. Two heuristic methods were evaluated. Heuristic *single intersection* inverse warps use only the first intersection of an inverse warp. For each pixel, the heuristic single warp continues warping until a confidence threshold is met. If no pixel meets the threshold, the result with the highest confidence is returned. No visibility calculations are performed. Heuristic *multiple intersection* warping uses a similar overall framework; however, it uses the inverse warp with multiple intersections. The results are combined using the visibility line technique described in Chapter 6. At any time, if the first visible intersection meets the confidence threshold, heuristic multiple intersection warping returns that result for the pixel. If all reference images have been considered, the first intersection is returned without regard to confidence. For the following examples, a confidence threshold of 0.5 was used.

I will refer to methods that always consider all images as complete methods. Two complete methods were considered. Complete single intersection warping considers only the first intersection from every reference image. Again, the visibility-line technique was used to solve the visibility problem, but, because no intervals are formed by the single intersections, visibility calculations are minimal. At the end of all warps, the first intersection that meets a minimum "skin" threshold is returned. In the following examples, intersections with confidences lower than 0.1[1] were considered skins. Finally, complete *multiple intersection* warping uses all intersections from all warps in the visibility line. For each pixel, the first intersection in each visibility line was returned as the final result, without consideration of its confidence.

---

[1] The confidence metric is scaled to 1. One is an exact area and squareness match. Zero is not an actually returnable confidence (a zero or infinite area would return a zero confidence).

Note the disparity quantization introduces some errors, most noticeable in multiple intersection heuristic and complete warping. This results in some “speckling,” single pixels missing in otherwise smooth surfaces.

7.2 Simple Scene

The first example scene is a sphere floating above a tiled surface. The left pair of each image is the result image. The right image is the result image with pixels coded in gray scale solely based on their originating reference image.

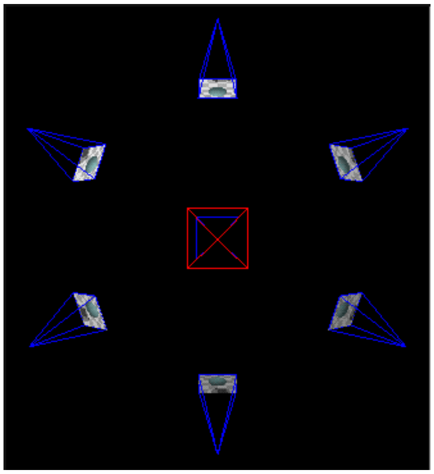

Figure 7.1: Scene View of the Simple Scene

This scene is probably atypical of the intended future applications of warping. The desired scene is a bird’s eye view, while the reference images are arrayed in a circle closer to the height of the sphere.

The center frustum is the desired image. The 6 reference images are spaced evenly around the sphere, with a slight downward angle. A sample reference image is shown, with the corresponding disparity map.

Figure 7.2: A Reference Image From the Simple Scene

The naive method is useful in giving an intuition into how the ranking heuristic performs. Note how the six images divide the desired image pixels in the right image.

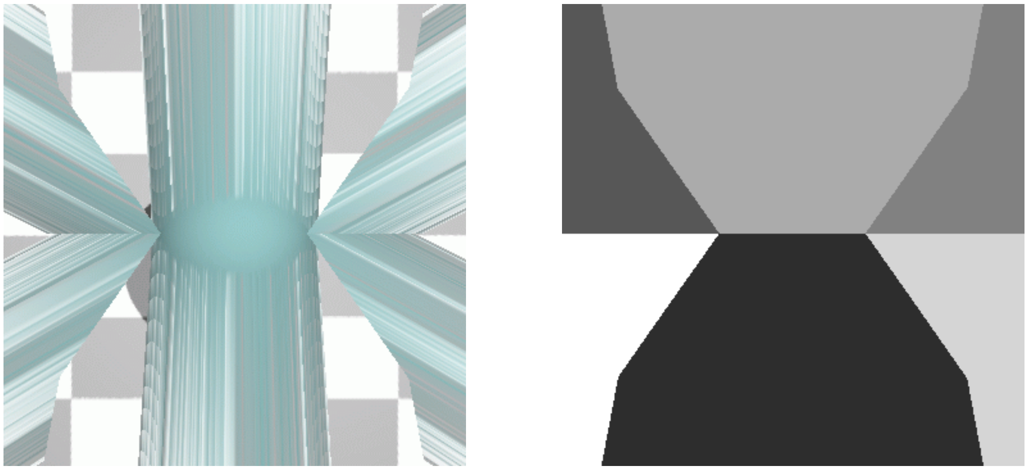

Figure 7.3: Naive Warp of Simple Scene

A minimum confidence threshold on single intersections immediately removes a lot of the skins. The mix of reference image choice visible, especially on the surface of the sphere shows the ability of inverse warping to use different reference images from pixel to pixel.

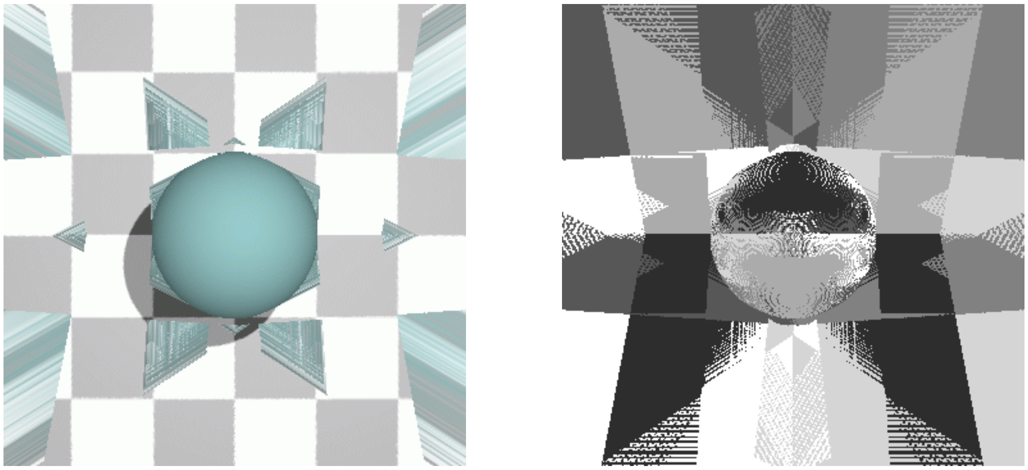

Figure 7.4: Heuristic Single Intersection Warp of Simple Scene

The multiple intersection warp gains little in this example. The “fuzziness” of the skins chosen at the top and bottom center is caused by the close ranking of two images in that area. When the reference image with the high confidence view of the plane is chosen first, we see the tiles of the floor. When the image with the lower confidence skin is chosen first, no other images have visibility information on that area, so the skin is returned.

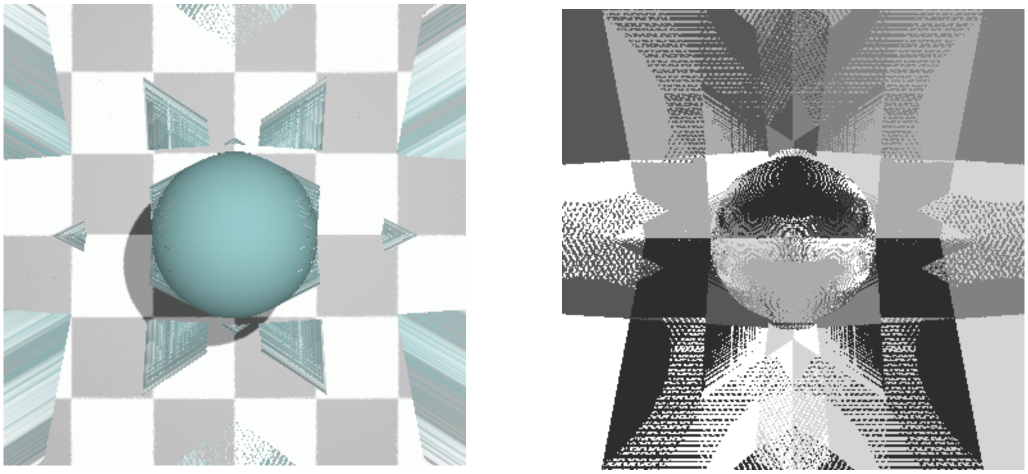

Figure 7.5: Heuristic Multiple Intersection Warp of Simple Scene

Complete single warping produces the best image for the simple case. The skin threshold allows removal of skins not possible in the other methods. Skins remain in areas where no reference image sees the tiled plane.

Figure 7.6: Complete Single Intersection Warp of Simple Scene

Unlike the heuristic multiple intersection result, complete multiple intersection warping returned skins at the top and bottom triangles at the middle of the image.

Figure 7.7: Complete Multiple Intersection Warp of Simple Scene

7.3 The Chair

The next scene is more indicative of a real scene. A chair is in the middle of the scene, with a teapot sitting on the chair. The back and seat of the chair are a fine mesh. Pay particular attention to the gap between the back of the chair and seat. This is an interesting area for visibility resolution.

Most of the reference images are arrayed at 45 degree angles around the teapot. They span 225 degrees. One reference image is a bird's eye view of the scene. The desired image is the frustum at the bottom of the scene view.

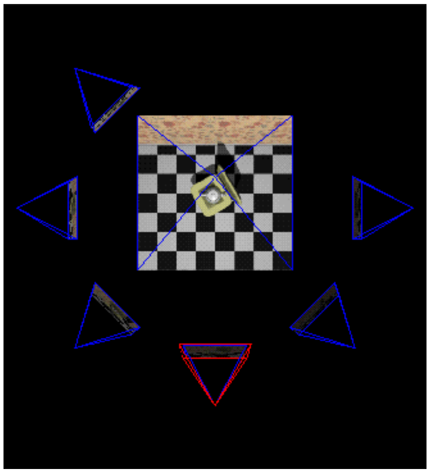

Figure 7.8: Scene View of the Chair

The images are slightly angled downwards and centered on the teapot The reference image and disparity map of the view at the farthest right in the scene view is provided.

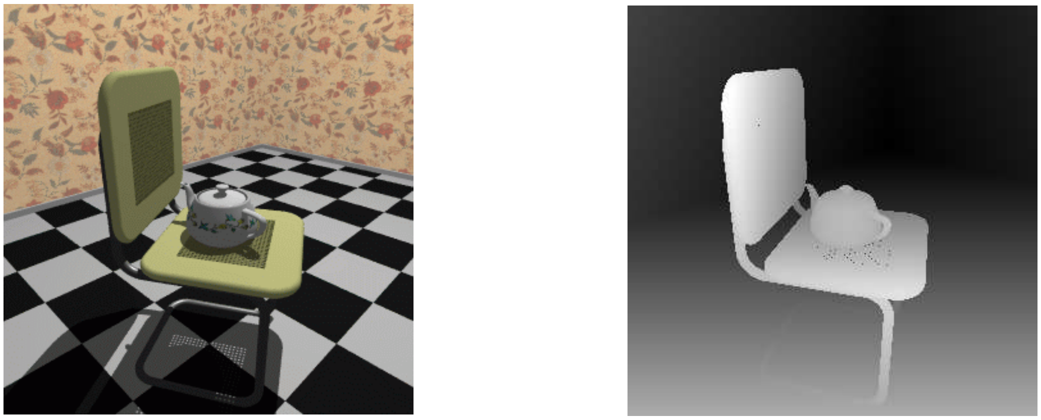

Figure 7.9: A Reference Image From the Chair

The naive method shows the initial partitioning of the desired image by the ranking method. Because ranking is purely based on the relative locations of reference images, the partitioning is very geometric.

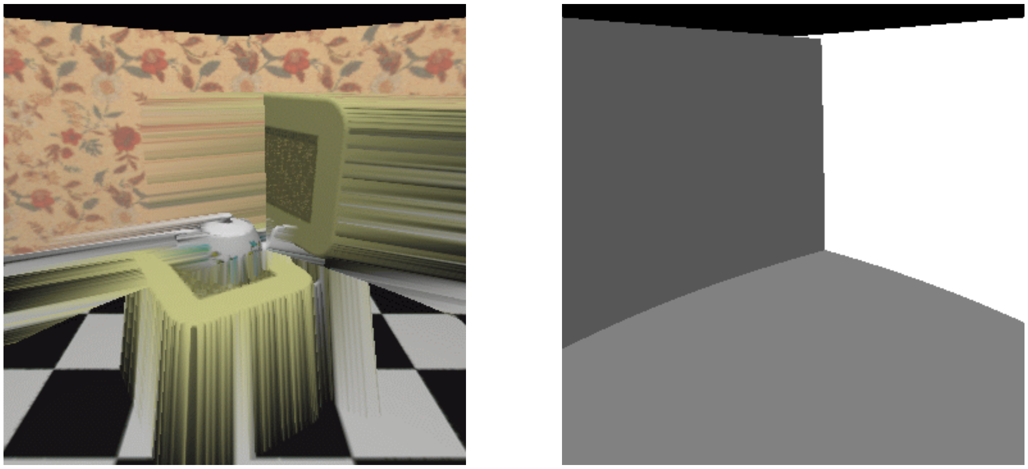

Figure 7.10: Naive Warp of Chair

Heuristic single intersection warping removes a lot of the obvious skins. The gap between the front and back of the chair is not clear, because the single intersection heuristic method performs no visibility calculations.

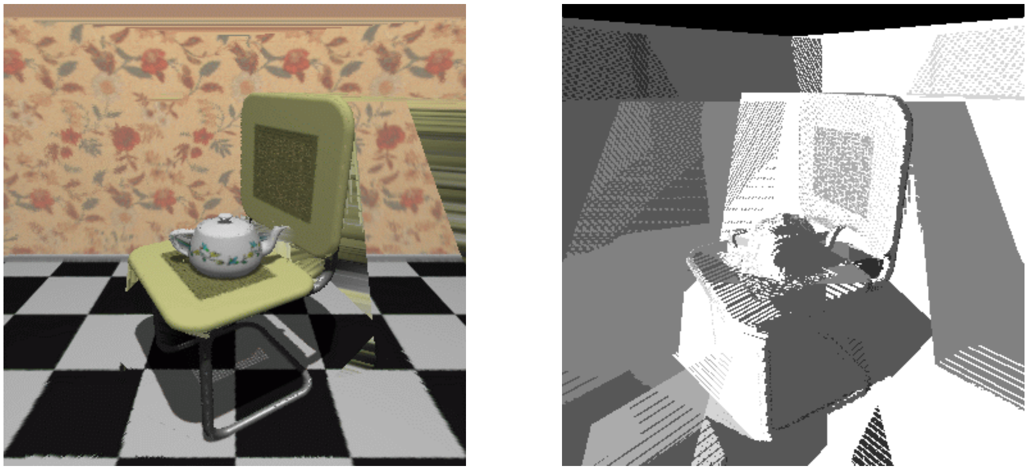

Figure 7.11: Heuristic Single Intersection Warp of Chair

The main gain of heuristic multiple intersection warping is the clearing of some skins through the gap in the chair. A skin on the back of the chair and at the right edge of the image remain.

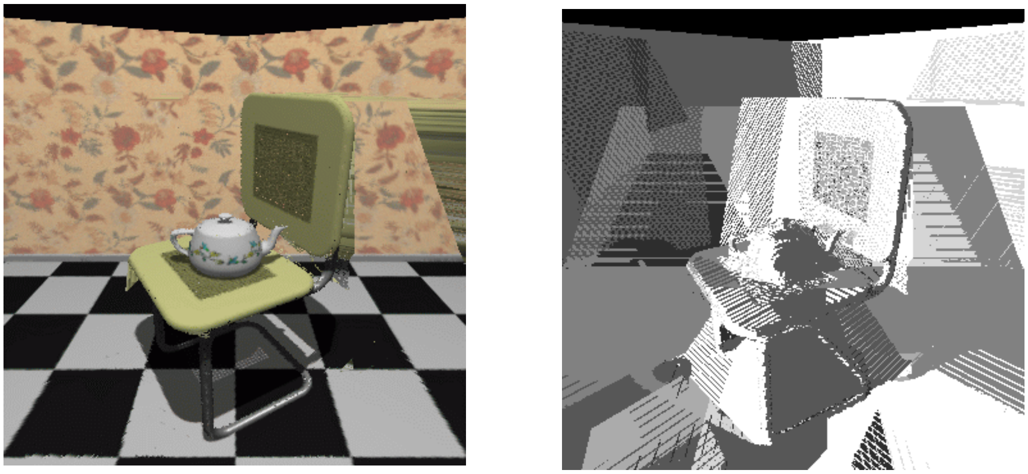

Figure 7.12: Heuristic Multiple Intersection Warp of Chair

Complete single intersection warping removes more of the skins on the left of the image. The simple visibility and skin thresholding still allows for the clearing of the gap of the chair.

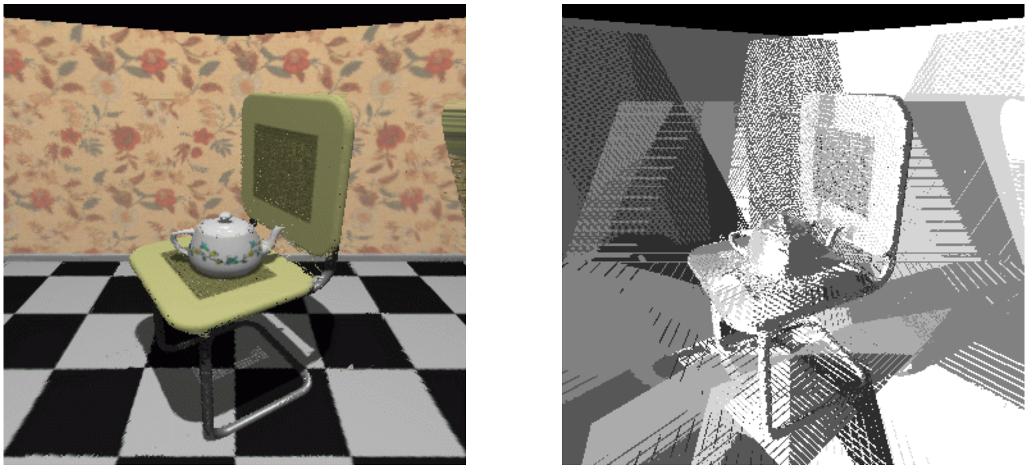

Figure 7.13: Complete Single Intersection Warp of Chair

Although the complete multiple intersection warp appears almost identical to the complete single intersection warp, the statistics and graphs in the next section show that the more comprehensive visibility calculations allowed higher confidence pixels to be found.

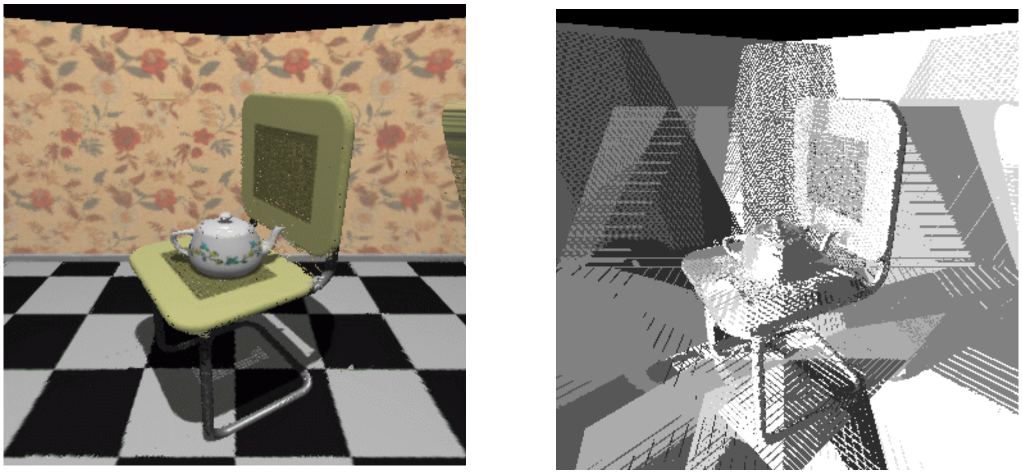

Figure 7.14: Complete Multiple Intersection Warp of Chair

7.4 Effectiveness

Statistics were gathered on the confidence of the resulting pixels from each method. Pixel confidence was recorded into 10 confidence ranges.

7.4.1 Simple Example Statistics

Table 7.1: Pixel Confidence in Simple Scene

Confidence Range	Naive	Heuristic Single	Heuristic Multiple	Complete Single	Complete Multiple
0.0 - 0.1	91583	30307	31144	20457	24424
0.1 - 0.2	1510	43	22	17	15
0.3 - 0.2	1647	63	57	55	55
0.4 - 0.3	13074	13570	11129	12644	11179
0.5 - 0.4	7431	20695	17150	23288	19264
0.6 - 0.5	1758	17799	17721	13029	9659
0.7 - 0.6	550	12714	12566	11227	9098

Table 7.1: Pixel Confidence in Simple Scene

Confidence Range	Naive	Heuristic Single	Heuristic Multiple	Complete Single	Complete Multiple
0.8 - 0.7	3104	10646	10986	15338	14890
0.9 - 0.8	5125	12015	13713	17903	20378
1.0 - 0.9	4178	12108	15472	16002	20998
Not Found	0	0	0	0	0
Above 0.5	14715	65282	70458	73499	75023
Below 0.5	115245	64678	59502	56461	54937

Some pixels in the simple scene only intersected skins, hence the large number of pixels at 0.1 confidence even in the complete methods. The more extensive number of intersections and corresponding visibility calculations allowed the multiple intersection versions of both heuristic and complete methods to find slightly higher confidence pixels than their single intersection counterparts. The complete methods also return more higher confidence pixels. Supporting the visual check supplied by the images in the simple example section, the confidence change from the heuristic to complete methods are incremental, while the naive method does poorly.

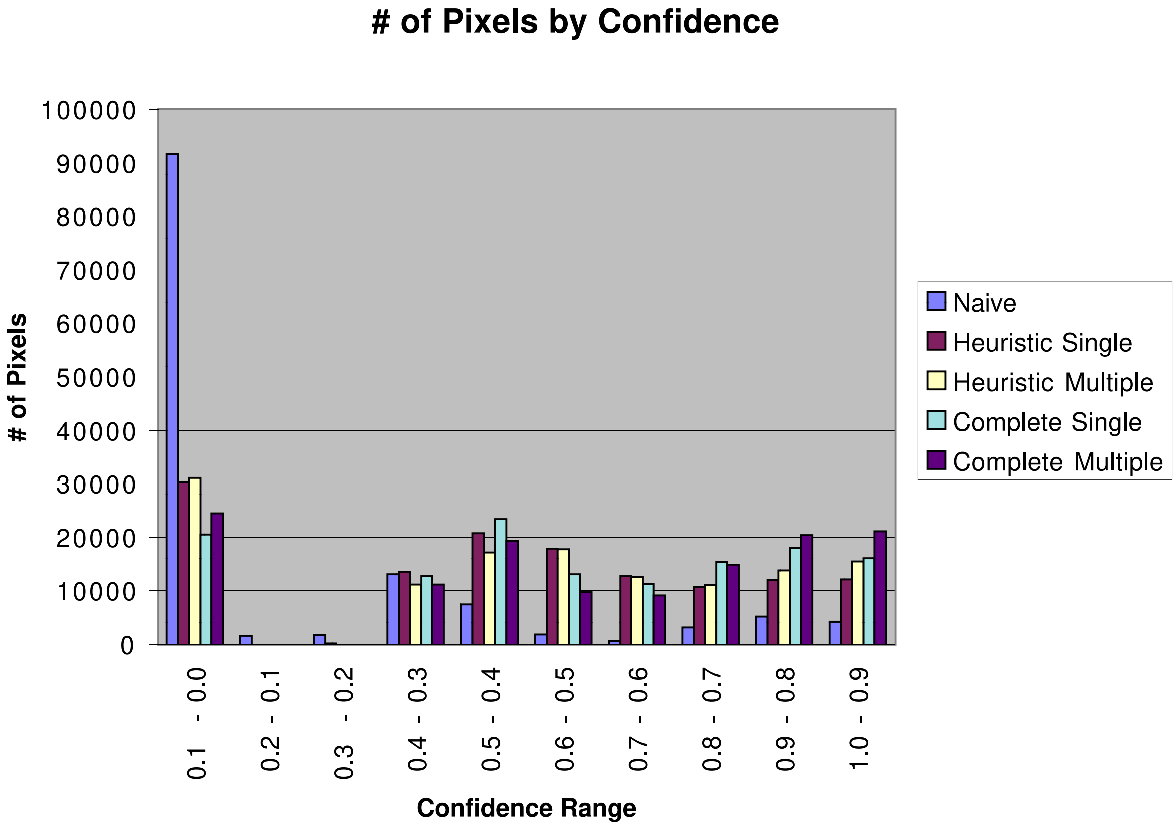

Figure 7.15: Pixel Confidence by Method in the Simple Scene

7.4.2 The Chair Statistics

Similar confidence breakdowns were gathered for the chair example.

Table 7.2: Pixel Confidence in the Chair

Confidence Range	Naive	Heuristic Single	Heuristic Multiple	Complete Single	Complete Multiple
0.0 - 0.1	58585	7730	4997	246	1707
0.1 - 0.2	4530	186	136	165	155
0.3 - 0.2	11825	6308	4822	7430	6304
0.4 - 0.3	11810	16952	14676	15390	15182
0.5 - 0.4	9424	17873	16351	17361	16614

Table 7.2: Pixel Confidence in the Chair

Confidence Range	Naive	Heuristic Single	Heuristic Multiple	Complete Single	Complete Multiple
0.6 - 0.5	8115	23234	25086	20229	18975
0.7 - 0.6	6077	17409	19319	18018	17144
0.8 - 0.7	5740	15955	17555	18817	19433
0.9 - 0.8	4654	12334	14253	16981	18484
1.0 - 0.9	2725	5504	6290	8848	9487
Not Found	6475	6475	6475	6475	6475
Above 0.5	27311	74436	82503	82893	83523
Below 0.5	96174	49049	40982	40592	39962

In this scene, the confidence of the pixels of the naive warp method scored even worse because of the large number of skins returned by the three reference images favored by the ranking heuristic. The heuristic methods find a significant number of pixels with confidence higher than 0.5, but the complete methods shift more of the pixels higher in the 1.0

range. Here, the slight gain of complete multiple warping is visible in the greater number of high confidence pixels versus even complete single intersection warping.

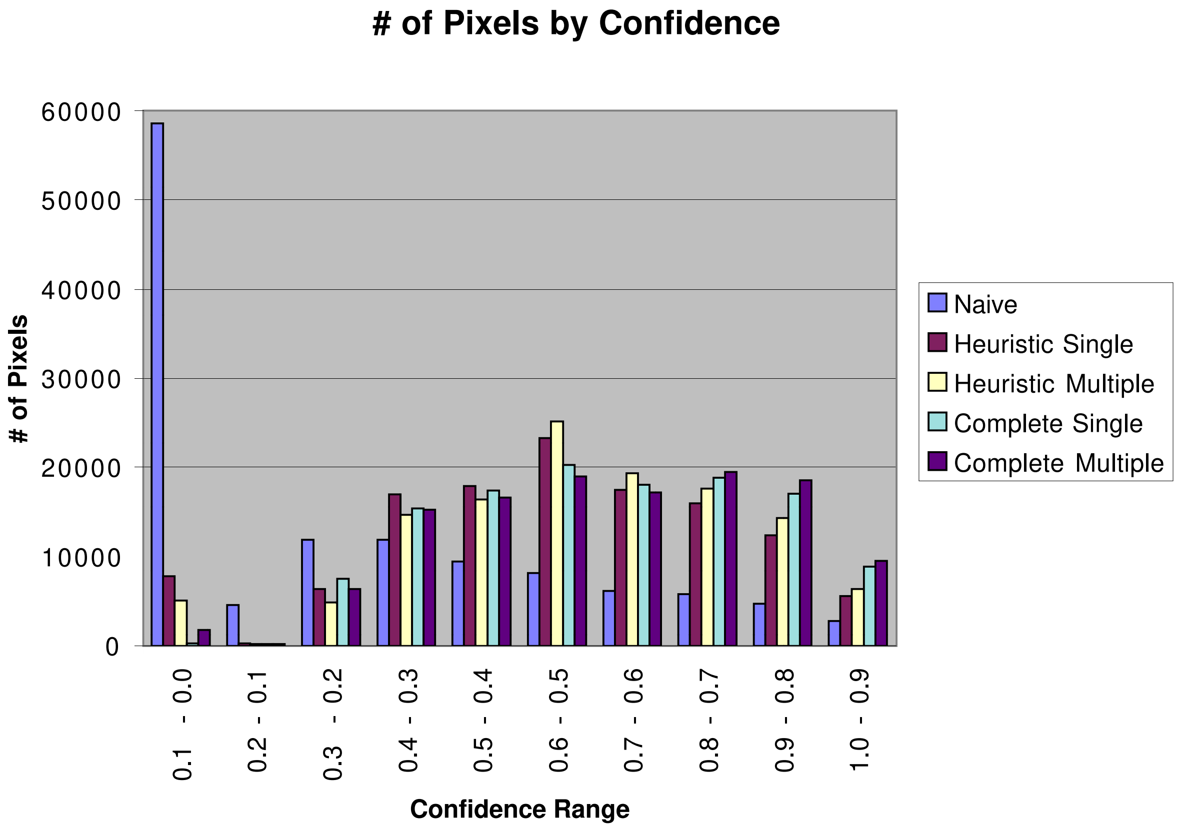

Figure 7.16: Pixel Confidence by Method in the Chair Scene

7.4.3 Discussion

Heuristic single intersection warping is a very cheap yet effective way to reduce the number of artifacts of inverse warping. However, its ability to produce pixels of adequate quality in areas of difficult visibility make its shortcomings obvious. Heuristic multiple intersections provides excellent answers at the cost of some correctness. Perhaps, most vexing, is when pixels on the border cases of the epipolar line length heuristic causes a ghostly skin (as in Figure 7.5). Complete single sacrifices some correctness to visibility for excellent results. Complete multiple warping provides the most valid visibility results but is computationally more expensive.

For these results, an optimized version of the base inverse warper was used. It included hierarchical epipolar line traversal as well as precomputation of bilinear surfaces [3]. Table 7.3 shows approximate running times for the five algorithms on the two different scenes. Although not rigorous the timing data provides some confirmation of the complexity of the different methods. The simple scene is not very indicative of performance because of its simplicity of geometry, but the chair scene provides some interesting visibility calculations. As expected, the methods with the more costly visibility calculations are much slower in the chair scene.

Table 7.3: Timings of Methods

Method	Simple	The Chair
Naive	9	9
Heuristic Single	19	21
Heuristic Multiple	36	59
Complete Single	46	94
Complete Multiple	53	110

Chapter 8

Conclusions

8.1 Future Work

The most intriguing aspect of inverse warping is its application to real images. Using cameras with depth finders or using images that have depth recovered from computer vision techniques, inverse warping is a perfect method for generating new images from real world scenes. The ability to capture scenes in an unconstrained manner sets it apart from many other image based rendering techniques. The best method for acquisition of these images and the relative merits of the different combination methods on reference images from the real world remains to be explored.

The visibility problem is a vexing one. One solution to the hidden occluder problem is to add the constraint that reference images must be cylindrical or panoramic in nature. Assuming a "normal scene," where most objects of interest lie more or less in a two dimensional plane, "tall" cylindrical images should work quite nicely. Some rethinking of the epipolar line length heuristic will be necessary. Another possibility is to use two heuristics in parallel. One ranks for pixel suitability; the other ranks for usefulness in visibility considerations. The exact nature of this metric and its calculation require further research. Like all heuristics, its effectiveness in use would need to be evaluated.

8.2 Contribution

I have explored the problems in combining the results of multiple reference images in inverse warping. I have suggested a method of ranking reference images before inverse warping to reduce the number of inverse warps required to find the best match for a desired pixel was developed. After warping, a confidence metric has been developed to measure the suitability of the resulting pixels. This confidence metric approximates the comparison of the solid angle of the desired pixel to the solid angle represented by the bilinear surface used to reconstruct the desired pixel. A visibility resolution scheme has been developed to use previous inverse warps to invalidate intersections that are artifacts of the reconstruction. By implementing and comparing several different methods, I tried to show the differences in image quality caused by the trade off of complexity and efficiency in dealing with these three issues.

References

[1] Gortler, S. J., R. Grzeszczuk, R. Szeliski, and M.F. Cohen, “The Lumigraph,” Computer Graphics (SIGGRAPH ‘96 Conference Proceedings), August 4-9, 1996, pp. 43-54.

[2] Levoy, M., Hanrahan, P., “Light Field Rendering,” Proceedings SIGGRAPH ‘96 (New Orleans, Louisiana, August 4-9, 1996). In *Computer Graphics* Proceedings, Annual Conference Series, 1996, ACM SIGGRAPH, pp. 31-42.

[3] Marcato, Robert., “Optimizing an Inverse Warper.” M. Eng. Thesis, Dept. of Electrical Engineering and Computer Science, MIT, 1998.

[4] McMillan, L., Bishop, G., “Plenoptic Modeling: An Image-Based Rendering System,” Proceedings SIGGRAPH ‘95 (Los Angeles, California, August 6-11, 1995). In *Computer Graphics* Proceedings, Annual Conference Series, 1995, ACM SIGGRAPH, pp.39-46.

[5] McMillan, L., “An Image-Based Approach to Three-Dimensional Computer Graphics,” Ph. D. Dissertation, Dept. of Computer Science, UNC, Technical Report No. TR97-013, 1997.

[6] Seitz, S. Dyer, C., “View Morphing,” Proceedings SIGGRAPH ‘96 (New Orleans, Louisiana, August 5-9, 1996). In Computer Graphics Proceedings, Annual Conference Series, 1996, ACM SIGGRAPH, pp. 21-30.