Real-Time Depth Warping For 3-D Scene Reconstruction

W. Brent Seales
Computer Science Dept
UNC Chapel Hill
Sitterson Hall CB #3175
Chapel Hill, NC 27599
seales@cs.unc.edu

Greg Welch
Computer Science Dept
UNC Chapel Hill
Sitterson Hall CB #3175
Chapel Hill, NC 27599
welch@cs.unc.edu

Christopher O. Jaynes
Computer Science Dept
University of Kentucky
773 Anderson Hall
Lexington, KY 40506
jaynes@cs.uky.edu

Abstract—This paper explores a new depth-recovery technique targeted for two environments: visualization on non-regular screen surfaces, and real-time depth recovery in an unknown scene. The algorithm is based on the computation of pre-warped images that are projected into the scene and re-imaged by synchronized cameras. The real-time control of the pre-warp and the type of imagery projected into the scene allows for a novel solution that recovers scene geometry and can be used in both visualization (projection of imagery onto screen surfaces) and computer vision environments.

TABLE OF CONTENTS

1. INTRODUCTION

This paper explores a new depth-recovery technique targeted for two environments: visualization on non-regular screen surfaces, and real-time depth recovery in an unknown scene. The algorithm is based on the computation of pre-warped images that are projected into the scene and re-imaged by synchronized cameras. The real-time control of the pre-warp and the type of imagery projected into the scene allows for a novel solution that recovers scene geometry and can be used in both visualization (projection of imagery onto screen surfaces) and computer vision environments.

Although there are many ways to compute depth from imagery [1], the algorithm we present principally combines the ideas of depth from structured light [2, 3] and depth from stereo vision [4, 5]. Specifically, structured light techniques project known images such as regular patterns (bars, grids) into an unknown scene and image that scene in order to recover scene geometry. Stereo reconstruction uses two or more views of the scene and corresponding points between those views to recover scene geometry. The work presented here uses projected imagery from a light source such as a Digital Light Projector (DLP) that forms a closed loop with a camera system that iteratively re-images the projected imagery, warps it, and re-projects it in order to compensate for distortion as a result of scene geometry. The iterative algorithm terminates when the warp has been computed to the pixel level, or when there is no other local warp that will better compensate for the distortion of the projected imagery as viewed by the camera system. Since the projected imagery, when re-captured by the synchronized cameras, exhibits warping solely due to depth variations in the scene, the captured image can be compared to the original projected image in order to solve for the depth geometry that induced the warp. The algorithm solves for the disparity-based warp using the stereo relationship of the projector and the cameras, and can be viewed as a dynamic structured light technique. The algorithm can potentially out-perform both stereo and structured light individually because it is iterative and has the advantage of gathering more images under adjustable conditions as it iterates and converges to a solution.

The next section reviews key concepts to provide a background to the details of the approach. The geometry of the camera system and the constraints that are exploited are presented in Section 3. An iterative algorithm to solve for the depth-induced warp is outlined in Section 4, and Sections 5 and 6 show visualization results from a simulator and summarize the primary results of this paper.

2. BACKGROUND

Conventional stereo reconstruction typically involves using two or more fixed cameras to simultaneously acquire overlapping 2D images of a scene. One image is then processed to identify interesting *features* such as intensity variations resulting from texture or structure in the scene. Next the overlapping portions of the remaining images are searched for corresponding features, i.e., features belonging to the same 3D scene point. Finally, using knowledge about the pose of the cameras and internal parameters such as camera focal lengths, rays are analytically projected from the center of each camera, through the corresponding 2D image feature, and into the scene. The intersection of these rays occurs at the 3D location of the point in the scene.

Geometric stereo is conceptually straightforward, yet in practice it can be a very difficult task. The problem of finding corresponding features in image pairs is confounded for example by ambiguous features, missing features (in

non-overlapping regions), and by object occlusions in the scene. Fortunately it is usually unnecessary to perform an unconstrained two-dimensional search of an image to find a feature match. Instead one can often make use of what is called the *epipolar constraint* [6, 7]. This constraint, discussed further in Section 3, is based on the fact that the true 3D location of the point in the scene must lie along the ray extending from the center of a camera, through the 2D image of the point, and into the scene. (See Fig. 1.) The crucial observation is that the image of the 3D point in a second camera must lie along the image of that ray in the second camera, the *epipolar line*. Thus the search for a feature match in a second image can be constrained to the one dimension along the feature's epipolar line. Finally, for convenience the camera image planes are often rectified to be coplanar and parallel to the baseline between the cameras. This can be done mechanically or in software. This rectification attempts to align the epipolar lines with rows of pixels in the images, further simplifying the search.

Because geometric stereo methods attempt to recover 3D scene structure passively, the quality of the results is directly related to the quantity and quality of the inherent "signal" in the scene, i.e., the identifiable features in the scene. Little or no texture in the scene can make correspondence difficult or impossible. To overcome this problem one can replace one of the cameras with a light projection device, e.g., a digital light projector, and actively "inject" texture or features into the scene. The results can then be imaged with any number of cameras. One can simply project random patterns to apply texture where there is none, or one can project *structured light* patterns such as vertical bars in a controlled manner to simplify the search for corresponding features in the camera images. For example, one could project binary coded vertical bars such as in [3]. In this example a sequence of n successive binary coded structured light patterns is projected into the scene, and a camera is used to observe the results. For each of the n camera images, the ON or OFF state of each camera pixel is determined by thresholding the measured intensity. The state of the pixel over the sequence of n images is used to construct an n-bit code identifying which bar was seen at each pixel. This correlation information, when combined with knowledge of the pose and internal parameters of the camera and projector, can be used to complete the triangulation to all scene points illuminated by the projector and seen by the camera.

Geometric stereo and structured light approaches to three-dimensional scene recovery are closely related in that both make use of knowledge about multiple cameras or projectors, along with correspondence between some set of features appearing in the respective images, to effectively triangulate the distance to those features in the scene. We present here a *hybrid* method that unifies these approaches in a closed-loop fashion, and enjoys the benefits of both. Like structured light methods our approach enjoys improved correspondence determination, yet like traditional stereo methods it is not intrusive or detrimental because we rely only on the imagery that was already being projected for the purpose of visualization (for example).

Furthermore, while the physical setup of our approach is geometrically the same as stereo vision, our approach is unique in that the projector is repeatedly projecting an every-day image into the scene, imaging the results, adjusting the image warp, and then re-projecting in a closed-loop fashion. In contrast, conventional stereo approaches attempt to recover the 3D scene structure with a single open-loop analysis of two or more simultaneously-acquired images. While the information in the conventional stereo scheme is limited to that contained in the simultaneous samples, our approach obtains additional information about the scene structure by effectively manipulating scene controls (warping the projected imagery) and directly observing the results. This closed-loop warp-project-image cycle is unique in that it allows us to converge on the correct 3D structure more rapidly and reliably than traditional methods.

3. GEOMETRY

Our algorithm requires that the relative geometry between the projector and the camera is known. Given the projection of a particular point from the projector to a point on the scene, our goal is to recover the depth of the scene at that point. This requires that each projected point is matched to a corresponding point as seen at the sensor. Corresponding points then are used to compute depth and the corresponding warp.

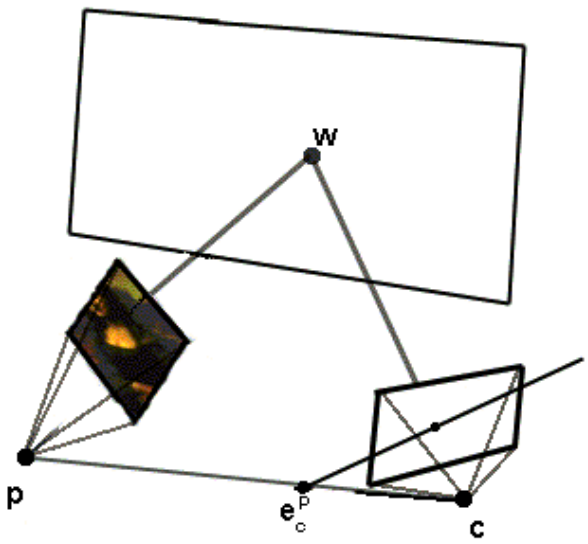


Figure 1: Epipolar geometry of the problem. A projector at position ***p*** generates a controlled image onto a surface of unknown shape. The known relative geometry of the camera and projector allows for the computation of epipolar lines in camera ***c***, for each projection ray that intersects the world. The image of ***w*** as sensed at ***c*** must lie along the epipolar line.

Our algorithm exploits the fundamental geometric constraint that can be derived from a system with two

known centers of projection. World points project onto the two image planes along a ray that intersects each center of projection. These two lines, and the known line between the two projection centers, referred to as the baseline, generate the epipolar plane. The image of the center of projection at ***p*** on the camera plane of ***c***, is the epipole, ***e***. An epipolar line in the camera plane at ***c***, can be computed as the camera plane with the epipolar plane. Therefore, for any point projected by ***p***, its corresponding image at ***c*** must lie along the appropriate epipolar line. This is shown in figure 1.

Given the known epipolar geometry, the computation of the depth of a particular point only requires a search along an epipolar line that emerges from the epipole and passes through the camera plane. Use of a constrained epipolar search for correspondences allows the depth of the world surfaces (for example, point ***w*** in Figure 1) to be computed from independent search along each of the epipolar lines. This implies a straightforward algorithm for the both the computation of the depth of points and the correct image pre-warp based on the recovered surface geometry.

4. ITERATIVE ALGORITHM

An iterative algorithm in the geometric context laid down in the previous section can capitalize on three important characteristics in order to solve for the depth-induced image warp. First, the projection device emits a known image. This means that the shift in features that occurs between the sensed image and the projected one can be detected more easily. Note that in stereo vision, for example, correspondence must be made between sets of images over which the system has no control – the images are sensed from the environment rather than projected into it. Second, the system is dynamic, allowing the projected image to be warped and re-projected very quickly and in a way that can be detected by the sensing device. Third, the relative geometry between the sensor and the projector is known. That is, the epipolar geometry is known and thus the search space for corresponding regions between the projected image and the image sensed by the camera is reduced.

The algorithm for computing the per-pixel depth warp operates on epipolar lines independently, which is geometrically correct when the image planes of the projector/camera pair are parallel, in the same plane, and when the optical centers lie in the same plane. Although this alignment is difficult to achieve with the physical devices, a rectification image transformation is easily computed as a pre-warp to the algorithm stated here.

Let I_p be the original image to be projected, and let I_c be the image that is sensed by the camera. The basis of the algorithm is to identify a feature in I_p and to find its position along the corresponding epipolar line in I_c. If the screen surface were at infinity, there would be zero disparity between the location of the feature in the two images. The shift that is detected is due to the depth of the screen surface along the optical rays defined by the camera, projector and pixel location of the feature. The task is to compute the warp to apply to the epipolar line of I_p on which the feature lies that will move the detected feature in I_c into the proper location. Thus an immediate approach is to warp the epipolar line I_p a small amount, project and image the result, and measure progress toward the correct location.

A more direct approach is to locate in I_c the position where the feature is desired. At this location, some arbitrary portion of the epipolar line from I_p is the correct correspondent, but finding that correspondent in I_p gives the exact location the feature should be warped to in order to guarantee it projects into I_c at the correct location.

Thus the algorithm to recursively compute the pixel-based warp for each epipolar line is the following:

Algorithm: Recursive Depth Warp

I_p: image projected through projector
I_c: image captured by sensor

For each epipolar line segment (treated independently for rectified images)

Locate a feature **f** in I_p

Assign feature **g** as the image region in I_c where **f** projects when there is zero disparity

Find the correspondent of **g** in I_p (correlation)

Create I'_p by warping **f** to **g** in I_p (warp the two segments of epipolar line linearly as **f** warps to **g**)

Re-project I'_p and recursively process sub-epipolar lines defined by **f** as the dividing point

This straightforward algorithm requires a correlation-based image matcher, rectified images, and does not include backtracking or explicit termination criteria. The algorithm terminates at pixel accuracy (each epipolar line is divided into subsections no longer than k pixels long) or when a total epipolar-line correlation measure is satisfied. Observing the quality of the **Find** step in the algorithm, which requires making a match based on a correlation measure, can monitor progress. An iteration of the algorithm corresponds to a locate-warp-project cycle, and therefore the algorithm requires k iterations for images with k pixels per scan line in order to reach single-pixel accuracy. Naturally multiple features per iteration can take place in the warp and we believe we can achieve quick convergence by performing more matches per iteration and using a scan-line based termination criterion rather than proceeding to the pixel level in every sub-interval.

5. RESULTS

We have performed initial image-based tests using a simulation environment based on 3D Studio Max software (Kinetix). The simulation environment allows construction of rectified camera geometry, the rendering of complex geometric environments, and the treatment of camera and projector models capable of projecting arbitrary imagery into a complex scene.

The results here show two image sets. The first set is illustrative of scenarios with complex imagery and simple surfaces. The second set of images shows a regular pattern that is projected into a scene with spherical and planar surfaces. The grid points show the warping and the compensation quite clearly in the face of more challenging surface geometry.

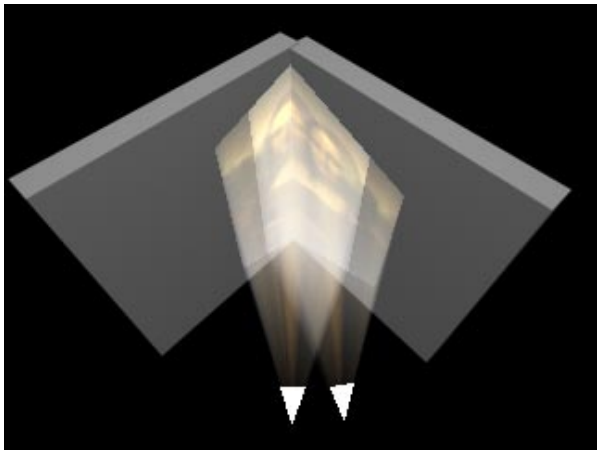


Figure 2: The simulation environment renders complex imagery using projector and camera models, which are interchangeable. Shown above are two camera models acting as projectors, emitting an image of the Mona Lisa. The image can be rendered into the environment and captured by any camera.

In most visualization environments, the display surface itself is featureless and regular, although its geometry is almost never pre-specified. The Mona Lisa has been selected as the imagery that is projected onto a simulated corner wall from an office [8] or a CAVE-like environment [9]. Figure 2 shows a view from above of the projector and the camera. In Fig. 2, both the camera and the projector are depicted as projectors in order to illustrate their position and the imagery they project/detect. The two image devices are rectified in the simulator such that scanlines are coincident with epipolar lines.

Figure 3 shows the first image projected by the projector and imaged by the camera. For reference, the projector is located on the left and the camera on the right as the view looks toward the display surface (Fig. 2). Notice that the

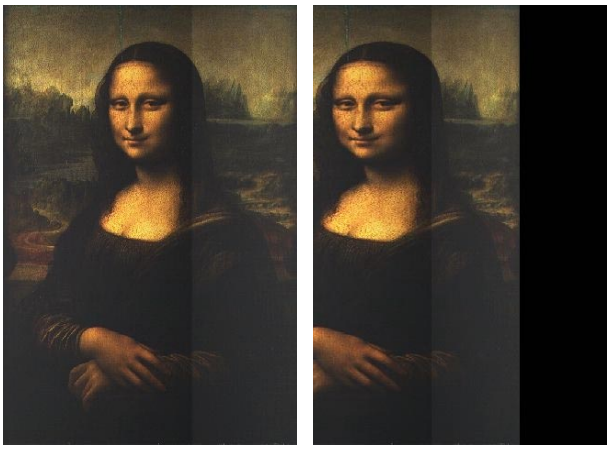


Figure 3: The left image is the projected image as viewed through the lens of the projector. The right image is the sensed image detected by the camera. Note the lateral warping that the display surface induces on the image seen by the camera. The eyes are 30% farther apart as a result of the projective warping caused by the tilted display surface.

sloping walls of the display surface induce a warp in the image viewed by the camera (Fig. 3, left). The algorithm isolates this warp and the goal becomes to warp the projected image in order to produce a version seen by the camera that is undistorted.

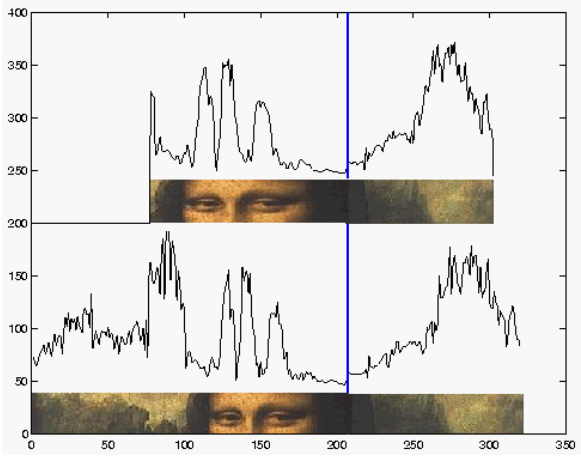


Figure 4: The intensity profile of a single scan line through the eyes compares the original projected image (bottom) to the warped version seen by the camera. The blue line locates the crease in the walls of the display surface.

In this example, the cameras are rectified and thus the problem is solved independently for epipolar (scan) lines. Figure 4 shows the intensity profiles of two scan lines from the image as seen by the projector (Fig. 3, left) and the warped image that is sensed by the camera (Fig 3, right). The intensity profiles in Fig. 4 clearly show the warp that spreads the eyes of the Mona Lisa apart due to the depth changes of the display surface. A correctly warped image, when projected, would produce an intensity profile that matches the original image.

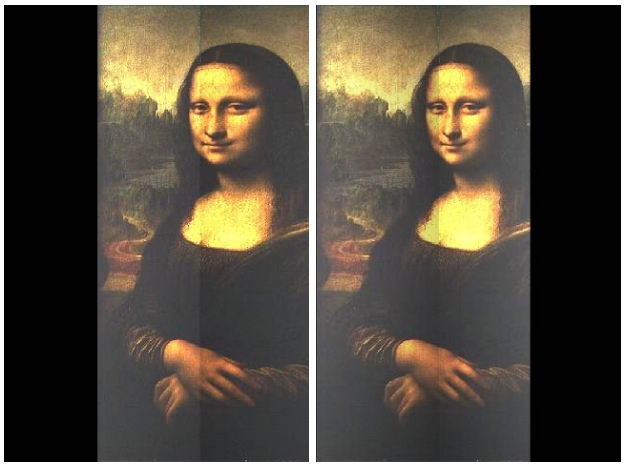


Figure 5: The left image is the warp that must be applied to the image before projection in order to produce the sensed image on the right, which matches exactly (except for a global translation that results from the baseline between the camera and the projector) the original imagery (see left image of Fig. 2)

Figure 5 shows the warp that correctly compensates for the depth distortion seen by the camera. The image on the left is the warped version of the original image that is re-projected through the projector and sensed by the camera. The camera captures a new image (Fig. 5, right) that is distortion-free when compared to the original imagery. Thus a viewer standing in the position where the camera is located would see correct imagery despite the depth variations of the display surface. The warp that is computed to generate the image to be projected is essentially a disparity map that corresponds to depth (when cameras are calibrated).

Fig. 6 shows the intensity profile of scan lines in the final images. The lower curve is the intensity profile of a scan line through the eyes of the original image. The upper curve is the same scan line from the image that is sensed by the camera *after* the correct depth-compensation warp is performed and the image is projected by the projector. Note that the profiles match almost exactly, with normal variation due to simulated optical and material effects (diffuse surfaces, ambient lighting, atmospheric attenuation, etc.)

A second image set illustrates the geometry when cameras are not "physically" rectified and when the projection screen geometry is more complex. A checker pattern was projected in the simulation environment onto a spherical/planar display surface. The non-rectified camera geometry implies that rectification must precede the application of the algorithm for computing the depth warp as it is stated here.

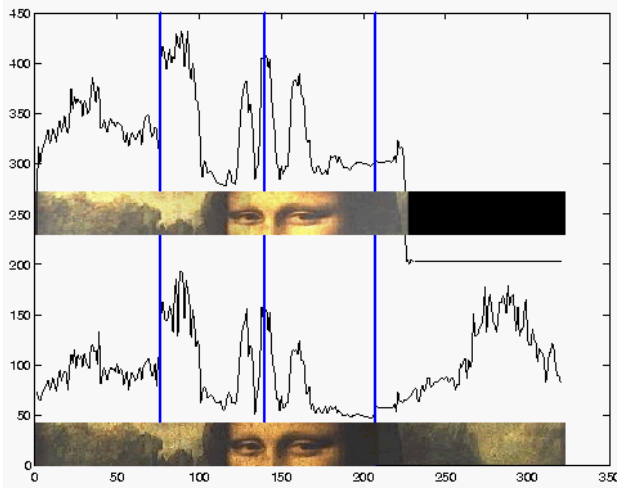


Figure 6: The intensity profile of a single scan line through the eyes compares the original projected image (bottom) to the version seen by the camera *after* the correct warp is found. Thus a correct solution means that the intensity profiles match: the camera captures an image exactly like the original. The correct pre-projection warp compensates depth-based distortion. The blue line locates corresponding image features.

Figure 7 (top) shows the view of the scene geometry from above, with the projector and camera (not shown) in the same configuration as before. The checkerboard pattern, when viewed from the same optical path as the projector itself, is regular an undistorted. The camera's view, however, clearly shows the depth-induced image warping that is the basis of the algorithm (Fig. 7, upper right).

6. SUMMARY

This paper presents an algorithm for recovering the depth of an unknown surface using active illumination. A second camera captures the imagery that actively illuminates the scene, and the captured image is compared to the original one. The warp between the two images is directly related to the relative position of the camera and projector as well as the structure of the scene. The algorithm exploits the possibility of dynamically modifying the projected imagery as the camera continuously observes it. This closed-loop formulation permits a recursive solution that terminates when the image warp is computed to the per-pixel level.

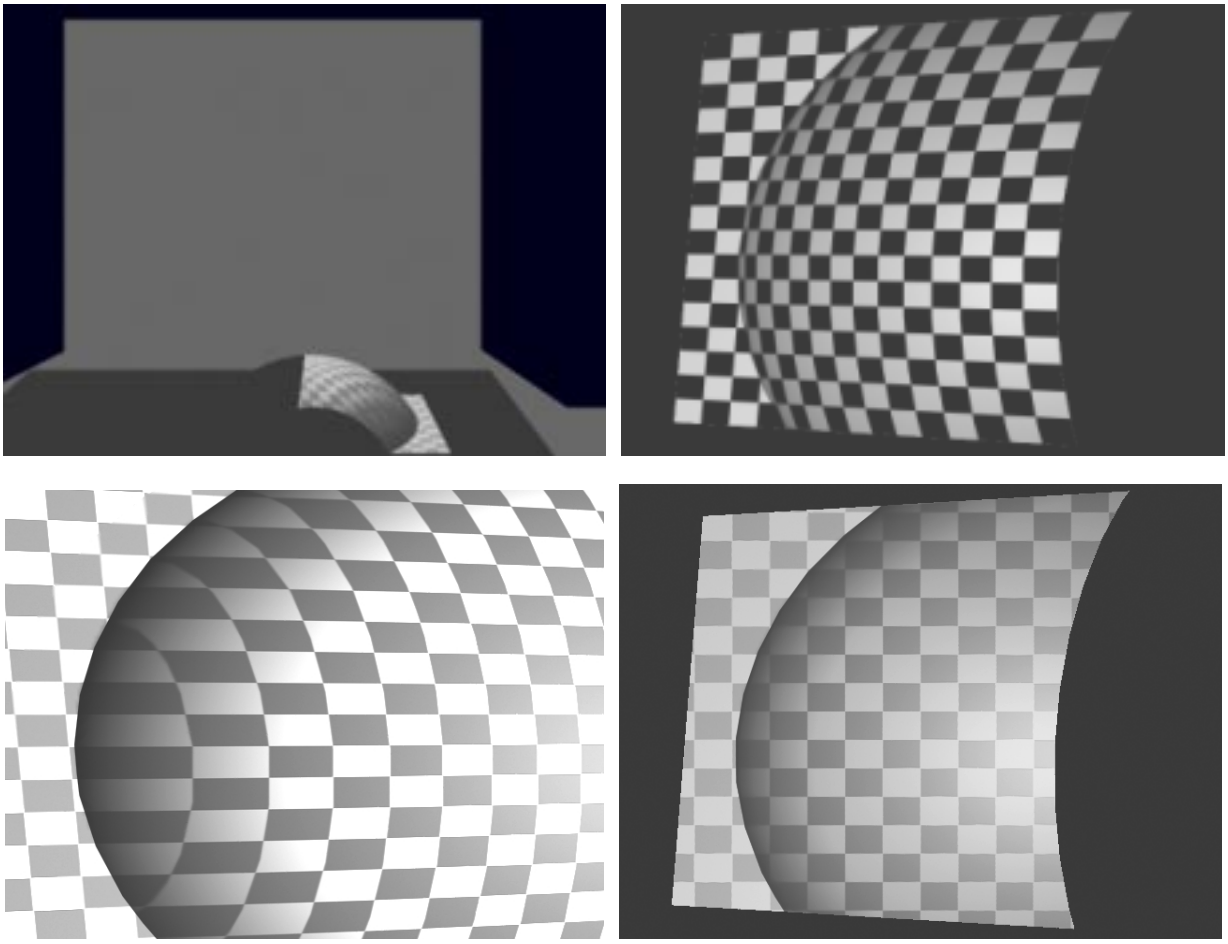


Figure 7: The spherical surface is shown from above (top left) to illustrate the relationship of the projector/camera rig with respect to the scene geometry. The camera's view of the projected pattern (top right) is distorted due to the projection surface. In this case the camera/projector pair are not rectified. The warp that is necessary to apply to the original image such that the projected imagery will be distortion-free is shown on the bottom left. This warped image, when projected and imaged by the camera, produces a correct image (bottom right).

We have shown that when the image planes of the camera and projector are rectified (image planes in the same plane, and line connection optical centers parallel to image plane) the problem of finding the per-pixel warp can be solved by processing individual scan lines independently. Processing scan lines independently allows the problem to be solved in parallel. When the camera/projector pair is not rectified a rectification warp can be computed.

The results from a simulation environment show that the warps can be computed with little or no information about relative camera position other than the rectification assumption. The immediate goal of this work is to build on the examples from the simulator by implementing the complete algorithm in a visualization environment such as the Office of the Future [8].

7. REFERENCES

[1] E. Krotkov, "Focusing." *International Journal on Computer Vision* Vol. 1 pp. 223-237, 1987.

[2] R.A. Jarvis. "A perspective on range-finding techniques for computer vision", *IEEE Trans. Pattern Analysis Mach. Intell*., 5:122-139, 1983.

[3] F. DePiero, M. Trivedi. "3-D Computer Vision Using Structured Light: Design, Calibration and Implementation Issues." Advances in Computers (43), pp.243-278, 1996.

[4] U.R. Dhond and J.K. Aggarwal, "Structure from Stereo-A Review." *IEEE Trans. Pattern Analysis Mach. Intell.,* 19(6):1489-1510, 1989.

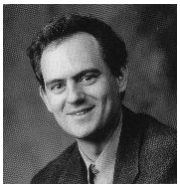
[5] O. Faugeras, (1992). "What can be seen in three dimensions with an uncalibrated stereo rig." Proc. Of the 2nd European Conference on Computer Vision, Italy.

[6] R. Deriche and O.D. Faugeras. "Tracking Line Segments." In *Proceedings of the 1st ECCV*, pages 259--268. Springer Verlag, April 1990.

[7] Keating, T. J., Wolf, P. R., and Scarpace, F. L. 1975. "An Improved Method of Digital Image Correlation," *Photogrammetric Engineering and Remote Sensing*, Vol. 41, No. 8 (August), pp. 993-1002.

[8] R. Raskar, G. Welch, M. Cutts, A. Lake, L. Stesin, and H. Fuchs, "The Office of the Future: A Unified Approach to Image-Based modeling and Spatially Immersive Displays." SIGGRAPH 1998, pp. 179-188.

[9] C. Cruz-Neira, D. Sandin, and T. DeFanti. "Surround-Screen Projection-Based Virtual Reality: The Design and Implementation of the CAVE." *Computer Grapics*, SIGGRAPH, 1993.



W. Brent Seales *holds the position of Associate Professor in the Computer Science Department at the University of Kentucky in Lexington, Kentucky. He received a BS from the University of Southwestern Louisiana, and an MS and PhD from the University of Wisconsin, Madison. All three degrees are in computer science.*

Greg Welch *holds the position of Research Assistant Professor in the Computer Science Department at the University of North Carolina at Chapel Hill. He received a BS in Electrical Technology from Purdue University, and an MS and PhD in Computer Science from the University of North Carolina at Chapel Hill. After leaving Purdue, prior to attending UNC, Welch worked at NASA's Jet Propulsion Laboratory and Northrop-Grumman's Electronic Systems Division.*

Chrisopher O. Jaynes *holds the position of Assistant Professor in the Computer Science Department at the University of Kentucky. He received a B.S. in Computer Science from the University of Utah, and a Ph.D. in Computer Science from the University of Massachusetts. Before joining the faculty at the University of Kentucky, Jaynes has worked as an Image Understanding Expert Consultant for the Sarnoff Research Center, and for Hewlett Packard as a software developer.*