Multiperspective Stereo Matching and Volumetric Reconstruction

Yuanyuan Ding Jingyi Yu
Department of Computer and Information Sciences
University of Delaware
Newark, DE 19716, USA
{ding,yu}@eecis.udel.edu

Peter Sturm
Perception Laboratory
INRIA Grenoble - Rhône-Alpes
Montbonnot 38334, France
Peter.Sturm@inrialpes.fr

Abstract

Stereo matching and volumetric reconstruction are the most explored 3D scene recovery techniques in computer vision. Many existing approaches assume perspective input images and use the epipolar constraint to reduce the search space and improve the accuracy. In this paper we present a novel framework that uses multi-perspective cameras for stereo matching and volumetric reconstruction. Our approach first decomposes a multi-perspective camera into piecewise primitive General Linear Cameras or GLCs [32]. A pair of GLCs in general do not satisfy the epipolar constraint. However, they still form a nearly stereo pair. We develop a new Graph-Cut-based algorithm to account for the slight vertical parallax using the GLC ray geometry. We show that the recovered pseudo disparity map conveys important depth cues analogous to perspective stereo matching. To more accurately reconstruct a 3D scene, we develop a new multi-perspective volumetric reconstruction method. We discretize the scene into voxels and apply the GLC back-projections to map the voxel onto each input multi-perspective camera. Finally, we apply the graph-cut algorithm to optimize the 3D embedded voxel graph. We demonstrate our algorithms on both synthetic and real multi-perspective cameras. Experimental results show that our methods are robust and reliable.

1. Introduction

Stereo matching is probably the most studied 3D scene reconstruction method in computer vision. Most existing approaches assume that the input cameras are perspective and tremendous efforts have been focused on handling issues unrelated to camera models such as textures, noise, specularity, and occlusion boundaries. Graph-cut based algorithms [4, 15, 16] and Belief Propagation (BP) methods [9, 30] can produce high accuracy and stable reconstructions. A common assumption in these methods is the existence of epipolar geometry [28]: the two images can be rectified to have a pure horizontal parallax. The epipolar geometry significantly reduces the search space and improves accuracy.

In this paper, we explore how to extend stereo matching to multi-perspective cameras. Recent developments have suggested that special types of multi-perspective cameras may also form valid stereo pairs. Seitz [28] and Pajdla [22] have independently classified all possible stereo pairs in terms of their epipolar geometry. They have shown that the epipolar geometry, if it exists, has to be a double ruled surface. Thus, very few varieties of multi-perspective stereo pairs exist. In general, a pair of multi-perspective cameras would not satisfy the epipolar constraint.

An alternative approach to stereo matching is volumetric reconstruction. Methods such as space carving [18] first discretize the scene into voxels and then prune the voxels based on their visibility and consistency with the input images. Level sets [7, 29] and graph-cut [3, 23] algorithms have been developed to simultaneously maintain reconstruction coherence and smoothness. An important step in volumetric reconstruction is back-projecting the voxels onto the input images. For multi-perspective cameras such as reflections on curved mirrors, this back-projection is highly non-linear and, in general, does not have a closed-form solution except for the most trivial cases.

We present a novel framework that uses multi-perspective cameras for stereo matching and volumetric reconstruction. Our method first decomposes each multi-perspective camera into piecewise primitive General Linear Cameras or GLCs [32]. GLCs uniformly model many existing multi-perspective cameras such as pushbroom and cross-slit cameras as 2D affine subspaces of rays. In this paper, we use GLCs as a tool for first-order approximation of arbitrary multi-perspective cameras.

Although a pair of GLCs generally do not satisfy the epipolar constraint, they still form a nearly stereo pair, i.e., the vertical parallax is nearly zero [6]. In this paper, we develop a new Graph-cut based algorithm to account for the slight vertical parallax using the GLC ray geometry. The recovered pseudo disparity map conveys important depth cues analogous to perspective stereo matching. To more accurately recover 3D scene geometry, we develop a novel multi-perspective volumetric reconstruction algorithm. We discretize the scene into voxels and apply GLC back-projections to map each voxel onto the input multi-perspective cameras. Finally, we apply the graph-cut algorithm to optimize the 3D embedded voxel graph. We demonstrate our algorithms on both synthetic and real multi-perspective cameras and show that our methods are reliable and robust.

2. Previous Work

Reconstructing 3D scenes from multiple views is probably one of the most explored problem in computer vision. Classical stereo matching algorithms use a pair of pinhole cameras to infer depth. The simple pinhole geometry allows any oblique pair of perspective images to be warped to have a purely horizontal parallax via projective transformations (homography) [11]. Efficient algorithms have also been developed for finding this transformation if the camera parameters are unknown [25]. We refer the readers to the stereo matching survey [26] for more details on existing methods.

Our goal is to apply stereo matching to cameras that do not follow pinhole geometry, i.e., they capture rays originating from different viewpoints in space. These cameras are often referred to as the multi-perspective cameras. Classical multiperspective cameras include pushbroom cameras [10], which collect rays along parallel planes from points swept along a linear trajectory, two-slit cameras [34], which collect all rays passing through two lines, and oblique cameras [22], in which each pair of rays are oblique. Despite their incongruity of view, some multiperspective cameras [24, 8, 22] can still form valid stereo pairs.

Seitz [28] classified all possible stereo pairs in terms of their epipolar geometry. Pajdla [22] independently obtained a similar result. Their results show that only three varieties of epipolar geometry exist: planes, hyperboloids, and hyperbolic-paraboloids, all corresponding to double ruled surfaces. Therefore, a pair of arbitrary multi-perspective cameras generally do not have valid epipolar geometry and hence classical stereo matching algorithms are not directly applicable to these cameras.

Another class of widely used 3D reconstruction algorithms are volumetric reconstruction [5, 27, 20, 2, 17]. Most existing algorithms in this category can be considered variations of the Space Carving framework by Kutulakos and Seitz [18]. Under this framework, an initial bounding volume is divided into a regular 3D voxel grid. Inconsistent voxels are then removed until the remaining voxels are photo-consistent with a set of input images. Similar to stereo matching, the input cameras are assumed to be perspective. This enables efficient voxel-pixel mapping when computing the voxel consistencies. For multi-perspective, computing this mapping is a challenging problem as closed-form solutions generally do not exist.

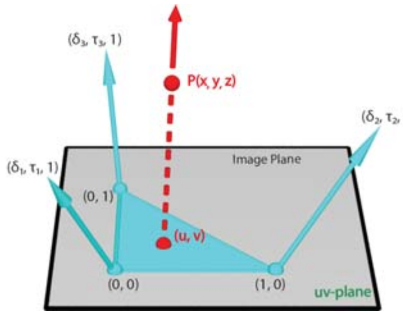

Figure 1. A General Linear Camera (GLC) collects rays that are affine combinations of three rays from pixel [0, 0], [0, 1], and [1, 0], respectively. A 3D point $\dot{P}$ can be uniquely mapped to a pixel in the GLC with closed-form solutions.

Our work is also related to catadioptric cameras and omni-directional vision. Catadioptric imaging systems place a pinhole camera at the focus of a hyperbolic or parabolic surface to synthesize a virtual pinhole camera with a wider field of view [21, 1]. This setup requires accurate alignment of the viewing camera [14, 31]. When the camera moves off the focus, the rays quickly diverge from a single perspective. Traditional catadioptric cameras, hence, have been restricted to mirrors with simple shapes whereas our framework applies to arbitrarily curved mirrors.

3. Multi-perspective Stereo Matching

A stereo pair consists of two images with a pure horizontal parallax, i.e., for every 3D point P, its images $[u_1, v_1]$ and $[u_2, v_2]$ in the two cameras must satisfy $v_1 = v_2$. Seitz [28] referred to this constraint as the stereo constraint and classified all possible (perspective and multi-perspective) stereo pairs in terms of their epipolar geometry. Conceptually, all rays in row v_1 of the first camera should intersect all rays in row v_2 in the second. Therefore, the epipolar surface should be double ruled. However, rays from a pair of multi-perspective cameras generally do not form a valid epipolar geometry. Therefore, we first introduce a new stereo model called the epsilon stereo constraint for multi-perspective cameras.

3.1. Epsilon Stereo Constraint

An epsilon stereo constraint allows slight vertical parallax between two cameras. We say that two views V and V' form an *epsilon stereo pair* if the following property holds:

The rays $V(u, v)$ and $V'(u', v')$ intersect only if $|v - v'| \leq \epsilon$.

The classical stereo constraint is a special case of the epsilon stereo model when $\epsilon = 0$. Any two such views are referred to as an ϵ-pair. Physically, an ϵ-pair represents two views with a mostly horizontal parallax and with a slight (ϵ) vertical parallax. In an ϵ-pair, all pixels on a row in one image correspond to pixels lying inside the ϵ band around the same row in the second, as shown in Figure 2. We also make the same assumption as [28] that all views are *u*- and *v-continuous* and only consider scene geometry visible in both views.

3.2. General Linear Cameras

We focus on applying the epsilon stereo analysis on a special class of multi-perspective cameras called the General Linear Cameras or GLCs [32]. In the GLC framework, each pixel corresponds to a ray and a GLC collects a 2D planar (affine) subspace of rays. Specifically, one can pick three generator rays and the GLC collects affine combinations of the three rays, as shown in Figure 1. The $st - uv$ light field parametrization are commonly used to parameterize the rays in GLCs.

In this paper, we simplify the GLC model by assuming the image plane (or the uv plane) coincides with the $z = 0$ plane. We then pick rays from pixels $[0, 0]$, $[1, 0]$, and $[0, 1]$ as the generator rays of the GLC. Assume the three rays have directions $[\sigma_1, \tau_1, 1]$, $[\sigma_2, \tau_2, 1]$, and $[\sigma_3, \tau_3, 1]$ respectively, we can then describe the direction $[\sigma(u, v), \tau(u, v), 1]$ of any ray at pixel $[u, v]$ as:

$$\begin{aligned}\sigma(u,v) &= (1-u-v)\cdot\sigma_1 + u\cdot\sigma_2 + v\cdot\sigma_3 \\ \tau(u,v) &= (1-u-v)\cdot\tau_1 + u\cdot\tau_2 + v\cdot\tau_3\end{aligned} \tag{1}$$

GLCs provide a unified description to most well-known multiperspective cameras, such as push-broom, cross-slit, and linear oblique cameras. Furthermore, since GLCs represent 2D planar ray manifolds, they provide a first order approximation of any multiperspective camera. We refer the readers to [32] for more detailed discussions on the GLC model.

Projection of a Point: The GLC has a closed-form projection from a 3D point $P[x, y, z]$ to pixel $[u, v]$ [33] as:

$$u = \frac{\begin{vmatrix} z\sigma_1 & z\tau_1 & 1 \\ x & y & 1 \\ z\sigma_3 & 1+z\tau_3 & 1 \end{vmatrix}}{Az^2+Bz+C}, \quad v = \frac{\begin{vmatrix} z\sigma_1 & z\tau_1 & 1 \\ 1+z\sigma_2 & z\tau_2 & 1 \\ x & y & 1 \end{vmatrix}}{Az^2+Bz+C} \tag{2}$$

where

$$A = \begin{vmatrix} \sigma_1 & \tau_1 & 1 \\ \sigma_2 & \tau_2 & 1 \\ \sigma_3 & \tau_3 & 1 \end{vmatrix}, B = \begin{vmatrix} \sigma_1 & v_1 & 1 \\ \sigma_2 & v_2 & 1 \\ \sigma_3 & v_3 & 1 \end{vmatrix} - \begin{vmatrix} \tau_1 & u_1 & 1 \\ \tau_2 & u_2 & 1 \\ \tau_3 & u_3 & 1 \end{vmatrix}, C = \begin{vmatrix} u_1 & v_1 & 1 \\ u_2 & v_2 & 1 \\ u_3 & v_3 & 1 \end{vmatrix}$$

Projection of a Line (Ray): GLC also provides a closed-form projection for a 3D line (ray) l. Assume l is not parallel to the uv plane, l will then intersect the uv plane at $[u_0, v_0, 0]$ and have direction $[\sigma_0, \tau_0, 1]$. All rays passing through l satisfy:

$$[u, v, 0] + \lambda_1[\sigma, \tau, 1] = [u_0, v_0, 0] + \lambda_2[\sigma_0, \tau_0, 1]$$

Eliminating λ_1 and λ_2, we have:

$$(u - u_0)(\tau - \tau_0) - (v - v_0)(\sigma - \sigma_0) = 0$$

We then replace σ, τ with u, v using Equation(1) as:

$$\begin{aligned}&(u - u_0)((1-u-v)\tau_1 + u\tau_2 + v\tau_3 - \tau_0) \\ &\quad -(v - v_0)((1-u-v)\sigma_1 + u\sigma_2 + v\sigma_3 - \sigma_0) = 0\end{aligned} \tag{3}$$

Thus l projects to a conic in the GLC.

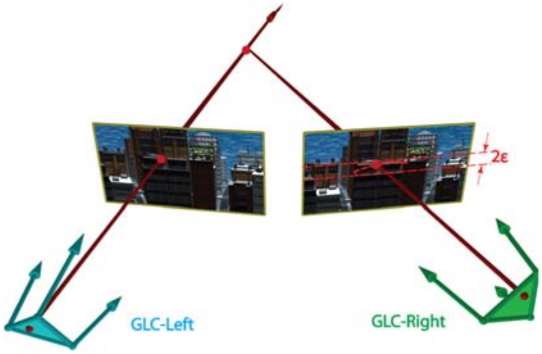

Figure 2. An epsilon stereo pair consists of two images with a mostly horizontal parallax and a slight (ϵ) vertical parallax.

3.3. Epsilon Stereo Analysis on GLCs

By Seitz's analysis, a pair of GLCs generally do not satisfy the epipolar constraint even if they are of the same type. For example, Feldman et al. proved that a pair of cross-slit cameras can have valid epipolar geometry only if they share a slit or the slits intersect in four pairwise distinct points [8].

However, a GLC pair will satisfy the epsilon stereo constraint. For example, consider the projection of a point $\dot{P}(x, y, z)$ in two GLCs of identical types but translated by $[-t_x, -t_y, 0]$. The image of $\dot{P}$ in the first GLC can be computed using Equation (2). To project P to the second GLC, we simply translate $\dot{P}$ by $[t_x, t_y, 0]$. The vertical and the horizontal parallax of $\dot{P}$ can be computed as:

$$\begin{aligned}\Delta v = v - v' &= -\frac{\begin{vmatrix} z\sigma_1 & z\tau_1 & 1 \\ 1+z\sigma_2 & z\tau_2 & 1 \\ t_x & t_y & 0 \end{vmatrix}}{Az^2+Bz+C} \\ &= \frac{z(t_y(\sigma_1-\sigma_2) + t_x(\tau_2-\tau_1)) - t_y}{Az^2+Bz+C}\end{aligned} \tag{4}$$

$$\begin{aligned}\Delta u = u - u' &= -\frac{\begin{vmatrix} z\sigma_1 & z\tau_1 & 1 \\ t_x & t_y & 0 \\ z\sigma_3 & 1+z\tau_3 & 1 \end{vmatrix}}{Az^2+Bz+C} \\ &= \frac{z(t_y(\sigma_3-\sigma_1) + t_x(\tau_1-\tau_3)) - t_x}{Az^2+Bz+C}\end{aligned} \tag{5}$$

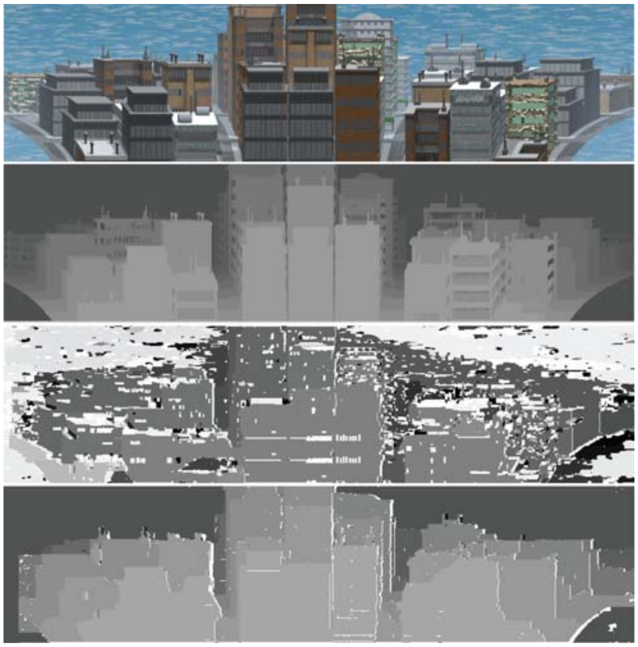

Figure 3. Epsilon Stereo Matching on Two Cross-slit Cameras. (a) shows one of the two cross-slit images. (b) is the ground truth depth map. (c) shows the recovered disparity map by treating the two images as a stereo pair and applying the graph-cut algorithm. (d) shows the horizontal disparity map recovered by our epsilon stereo mapping algorithm.

where A, B, and C can be computed by Equation (3).

Notice that the vertical parallax Δv is simply a quadratic rational function of z. In [6], it is shown that Δv is bounded and, hence, the two GLCs form an epsilon stereo pair. A similar analysis applies to any pair of GLCs [6].

3.4. Epsilon Stereo Graph Cut

A brute-force approach to use the epsilon stereo constraint is to modify classical stereo matching such as Graph-Cut or Belief Propagation by extending the disparity label set to two dimensions. Recall that all pixels in a row in one camera map to an $\pm\epsilon$ band around the same row in the other camera, where ϵ is bounded and usually very small. Therefore, we can simply use a 2D search space in standard stereo matching. The downside, however, is that the corresponding rays under such a labeling can be arbitrarily oblique and do not map to a 3D point.

To enforce coherent epipolar geometry, we add an additional penalty term to measure the "closeness" between two rays. Specifically, for each pixel $p[u_L, v_L]$ in the reference camera GLC_L, we first compute its corresponding ray $r_p = [\sigma_L, \tau_L, u_L, v_L]$ from p, where σ_L, τ_L can be computed using Equation (1). We then project ray r_p in the second camera GLC_R. Recall that the image of r_p in GLC_R is a conic curve $\mathcal{C}_p$ as shown in Equation (3). Therefore, to measure how well a pixel $q[u_R, v_R]$ in GLC_R matches $p[u_L, v_L]$, we measure both the color consistency between p and q and the distance from q to $\mathcal{C}_p$. Notice that if the corresponding rays from p and q intersect in 3D space, q should lie on $\mathcal{C}_p$ and the distance should be zero. In our implementation, we compute the distance by finding the closest point on $\mathcal{C}_p$ to q and we call it reprojection distance $D_{reproj}(p, q)$.

Finally, we formulate the epsilon stereo matching problem as a 2D disparity labeling problem. The search domain for each pixel p in GLC_L is defined as a window of height 2ϵ and width d^x_{max} in GLC_R, where d^x_{max} is the maximum horizontal disparity and ϵ is computed from the GLCs parameters. We define the energy function for a specific labeling f as:

$$E(f) = E_d(f) + E_{occ}(f) + E_s(f) + E_{reproj}(f) \quad (6)$$

The first three terms are commonly used in classical stereo matching: a data term E_d to measure color consistency, a occlusion term E_{occ}, and a smoothness term E_s [15]. The last term measures the reprojection error term E_{reproj}, i.e.,

$$E_{reproj}(d_x, d_y) = \sum_u \sum_v \mathcal{D}_{reproj}(p, p + [dx, dy]) \quad (7)$$

It is also important to note that although the final disparity map cannot be directly translated to a depth map of the scene, this pseudo disparity map provides useful depth cues analogous to perspective stereo matching. In Figure 3(d), we plot the recovered horizontal disparity and compare it with the ground truth depth map Figure 3(b).

4. Multi-perspective Volumetric Reconstruction

Next, we show how to use multi-perspective cameras for volumetric reconstruction. Most existing algorithms in this category can be considered variations of the foundational framework by Kutulakos and Seitz [18], in which a set of N perspective input images are used to determine 3D volumetric scene geometry. These methods, however, are not directly applicable to multi-perspective cameras due to non-uniform sampling and non-central projections.

4.1. Problem Definition

Given a set of N multi-perspective views to the scene, we want to reconstruct a volumetric representation of the scene. A typical scenario is to position multiple curved mirrors in the scene and then capture an image of these mirrors reflecting towards the scene, as shown in Figure 4.

In classical volumetric reconstruction, the scene is first discretized into voxels of size coherent with the input image resolution. For multi-perspective cameras, non-uniform sampling and multi-perspective distortions may lead to uneven discretization. Therefore, we first position a virtual

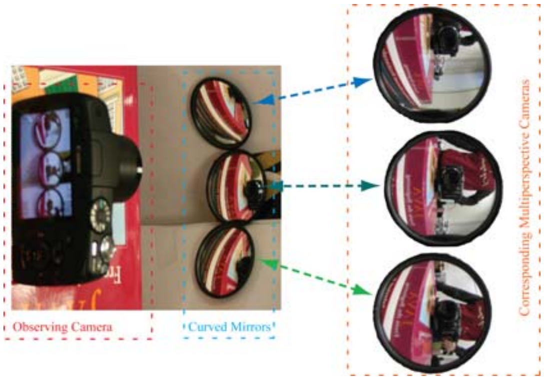

Figure 4. Our multi-perspective volumetric reconstruction recovers 3D scene geometry from multiple multi-perspective camera images.

pinhole camera K_v between the multi-perspective cameras and then uniformly discretize its viewing frustum volume in the projective space as shown in Figure 6. One can then prune the voxels based on their consistency with the input images and visibility.

To measure the color-consistency, we need to first determine the corresponding image of the voxel in each input camera. For perspective cameras, we can simply project the voxel into the camera. For multi-perspective cameras such as reflection on curved mirrors, computing this back-projection is a typical inverse ray-tracing problem and, in general, does not have closed-form solutions.

4.2. Multi-perspective Back-Projection

To resolve the back-projection problem, we first decompose each multi-perspective camera into piecewise GLCs. To do so, we tessellate a multi-perspective image into triangles and then find the associated ray at each vertex. Finally, we treat the ray triplet from each triangle as a GLC and use the triangle plane is the default uv plane.

The advantage of using this approximation is that GLCs provide closed-form projections. Most 3D points have unique projections in a GLC (except for the singularity points [33]) whereas other multi-perspective decompositions [13] have multiple projections from a 3D point.

The simplest approach to use the GLC decomposition for back-projecting a voxel is to go through every GLC in the tessellation and compute the image of the voxel using Equation (2). The search stops when the projected voxel lies inside the GLC triangle. However, controlling the GLC tessellation level can be challenging: a fine tessellation produces more accurate approximation but also requires more computations.

We therefore develop a dynamic tessellation scheme similar to the level-of-detail (LOD) technique in computer graphics. We first tessellate the multi-perspective camera using a coarse set of GLCs and then perform standard 1-to-4 subdivision and store the subdivision in a quad tree as shown in Figure 5. To back-project a 3D point $\dot{P}$ to the camera, we start with the top level GLCs and compute the image of $\dot{P}$'s projection. We determine which GLC contains the final projection and repeat our search on its children GLCs. The search stops until we reach the leave nodes. In our experiment, $4 \sim 5$ subdivision levels are usually sufficient to accurately back-project a voxel on complex multi-perspective cameras. The detailed back-projection steps are shown in Algorithm 4.1.

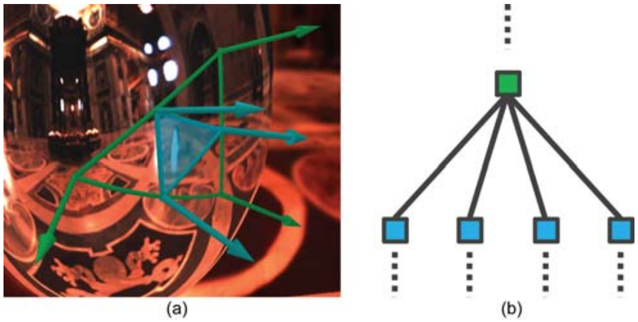

Figure 5. Multi-perspective Decomposition Using the GLCs. We can decompose an arbitrary multi-perspective camera (e.g., reflections on a curved mirror) using piecewise GLCs. We create a multi-resolution decomposition using subdivision and store it in a quad-tree.

Algorithm 4.1: MIRROR_GLCBACKPROJECTION($glc, \dot{P}$)

```
procedure GETRAY($const$ GLC &$glc$, $const$ Point3D &$\dot{P}$)
  $p[u,v] \leftarrow glc$.Project($\dot{P}$);
  if $p[u,v] \notin glc.triangle$
    then {return ($false$)
  if isLeaf($glc$)
    then {return ($p[u,v]$)
    else {
      $bNotFind \leftarrow true$;
      while $bNotFind$
        do {
          $x \leftarrow glc.subGLCs$.getNext();
          $q[u_1,v_1] \leftarrow x$.Project($\dot{P}$);
          if $q[u_1,v_1] \in x.triangle$
            then {$bNotFind \leftarrow false$;
      return (GetRay($x, \dot{P}$))
```

4.3. Graph-cut Optimization

Finally we apply volumetric reconstruction by first initializing a sweeping planes parallel to the image plane of the virtual perspective camera. We sweep the planes from distance d_{min} from the COP of the camera at interval d_i until we reach a pre-defined distance d_{max}.

At every distance value d_i, we find the voxel $\dot{P}_i(x,y)$that corresponds to pixel $[x,y]$ in the virtual perspective camera K_v. We then back-project the voxel to every multi-

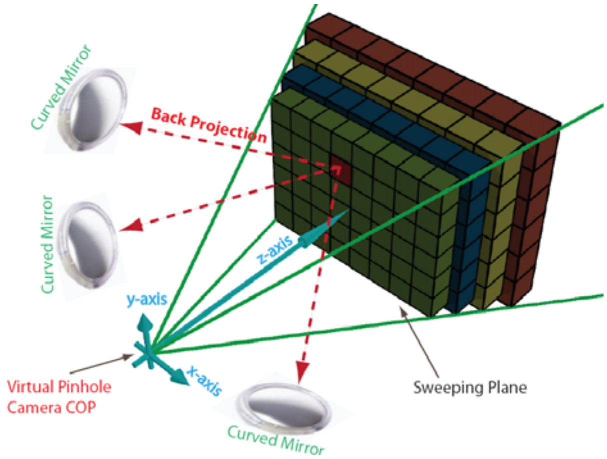

Figure 6. We construct the voxel space as the viewing frustum of a virtual perspective camera. We then back-project the voxels to all multi-perspective viewing cameras.

perspective camera $\mathsf{K}_j, (j = 1, \cdots, N)$ using the GLC-based back-projection algorithm. Assume the corresponding pixel of $\dot{P}_i(x, y)$ in camera j has color $[r_{ij}, g_{ij}, b_{ij}]$, $j = 1, \cdots, N$, we then measure the color consistency of these pixels in terms of their variance $m(x, y, d_i)$.

To reconstruct the volume, we can apply any existing volumetric reconstruction method such as space carving to compute the optimal volumetric 3D scene model. In our implementation, we have adopted a simplified version of [23]. We assume the 3D scene is a Monge function $z = f(x, y)$ with respect to the virtual perspective camera and we model the problem of finding the optimal z as a labeling problem and solve it using the graph-cut algorithm. Specifically, we assign the depth labels f to all pixel. Notice that a good labeling scheme should not only be consistent with the observed images but also smooth. Therefore, we set out to find the optimal labeling that minimizes the energy function:

$$\mathcal{M}(f) = \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} m(x_i, y_j, f(x_i, y_j)) \Delta x \Delta y \tag{8}$$

$$\begin{aligned} \mathcal{S}(f) = & \sum_{i=1}^{n_x - 1} \sum_{j=1}^{n_y} |f(x_{i+1}, y_j) - f(x_i, y_j)| \Delta y \\ & + \sum_{i=1}^{n_x} \sum_{j=1}^{n_y - 1} |f(x_i, y_{j+1}) - f(x_i, y_j)| \Delta x \end{aligned} \tag{9}$$

where $\mathcal{M}(f)$ encodes the consistency measure and $\mathcal{S}(f)$ encodes the smoothness measure. Finally, we find the optimal labeling via the standard graph-cut algorithm.

5. Experimental Results

We have validated our framework on both synthetic and real data. All experiments are conducted on a PC with 2.4 GHz intel Q6600 CPU and 3.0GB memory. For synthetic data, we use the POV-Ray Ray Tracer to render two cross-slit panoramas of resolution 1200x300 of the city scene. The two cameras we choose do not share a common slit or satisfy the stereo condition by Feldman et al. [8]. Figure 3 shows the ground truth depth map with respect to the image plane. Next, we directly apply a graph-cut stereo algorithm as if the two cameras were perspective. The resulting disparity map is highly noisy due to the violation of the epipolar constraint, as shown in Figure 3(c). Finally, we show our epsilon stereo matching result using the multi-perspective stereo matching algorithm. To minimize computational overhead, we pre-compute the conic curve image of each ray from the reference camera. We rasterize these curves and store the pixels covered by each curve for easy query. We compute the epsilon value of 12 pixels from the epsilon stereo analysis. In Figure 3(d), we plot the recovered horizontal epsilon disparity map. It takes 40 seconds for our algorithm to complete the estimation. Notice that although this map is not equivalent to the actual depth map, it is highly coherent with the actual depth map and reveals useful depth cues.

For volumetric reconstruction, we first use POV-Ray to render a perspective image viewing two spherical mirrors of radius 1 at a resolution of 1024x1024 (Figure 7(a)). We then position multiple objects at different depths within range 5 and 20 from the mirrors. We use a virtual perspective camera between the two mirrors reconstructing the 3D volume. We discretize the volume into 800x800x60 voxels, where the z depth is sampled between $d_{min} = 3$ to $d_{max} = 33$ at 0.5 interval. We use the GLCs to decompose each mirror into 6 triangles in the first level and apply standard 1-to-4 subdivision 4 times. Finally, we apply our multi-perspective volumetric reconstruction algorithm to recover scene depth. In Figure 8, we show our algorithm is able to accurately recover the contour of multiple scene objects (e.g., the "ICCV" sign) at different depths. Notice that the recovered depth is also horizontally flipped from the mirror image due to reflections.

For real data, we have constructed a spherical mirror array and experimented our volumetric reconstruction algorithm on a small subset of mirrors. Our array contains 19 identical mirrors of radius 89.3mm. We tightly pack the mirrors on a flat panel and apply similar processes in [19] to calibrate the relative positions between mirrors. We place a Canon SX100 digital camera approximately 90cm away from the mirror to capture the image of 7 mirrors at a resolution of 3456x2592. We pre-calibrate the viewing camera and use the red-dot pattern printed on the mirror panel to determine the camera pose. In Figure 8, we slightly tilt the camera to avoid self-occlusions in the reflection. We position the virtual perspective camera at the central mirror with a field-of-view of 80 degrees (approximately the same as central mirror) The scene objects lie at about 70cm away

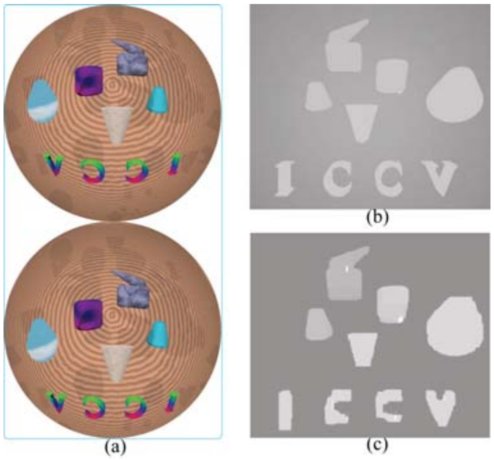

Figure 7. Recovering Synthetic Scenes. We use PovRay to render reflections on two spherical mirror in a synthetic scene. (a) shows the two mirror images. (b) shows the ground truth depth map and (c) shows our recovered depth map.

from the mirror array, we discretize the volume by 400x400 voxels in xy dimension and 90 voxels in z direction between $d_{min} = 10cm$ and $d_{max} = 100cm$. Columns (a) and (b) in Figure 8 illustrate the images observed on the central mirror and the upper mirror and Column (d) shows the recovered depth (from the volume) of the scene. To illustrating the quality of the recovered depth map, we use forward mapping to warp all reflection rays from the central mirror to a plane in 3D space to approximate a virtual perspective view as shown in column (c).

In Figure 9, we apply our framework for recovering a more complex scene. The viewing camera captures an image of itself as shown in Figure 9(a). We reconstruct the viewing frustum volume of a virtual perspective camera positioned near the center mirror, with similar field-of-view, range, and resolution as in the previous example. Despite the complexity of the scene, our method is still able to recover a reasonable depth map as shown in Figure 9(c). In our experiments, we also find that if the camera's viewing direction is perpendicular to the mirror array, the parallax between the neighboring mirror images becomes less discernable, i.e., the depth difference between neighboring layers would be difficult to detect. For example, the Canon camera is positioned close to the background checkerboard and our method is not able to recover its correct depth.

6. Conclusions and Future Work

We have presented a unified framework for multi-perspective stereo matching and volumetric reconstruction using the General Linear Cameras (GLCs). For stereo matching, we have introduced an epsilon stereo model to model a pair of GLCs that do not satisfy the stereo constraint. We have then developed a new graph-cut algorithm for finding the optimal epsilon disparity map. We have

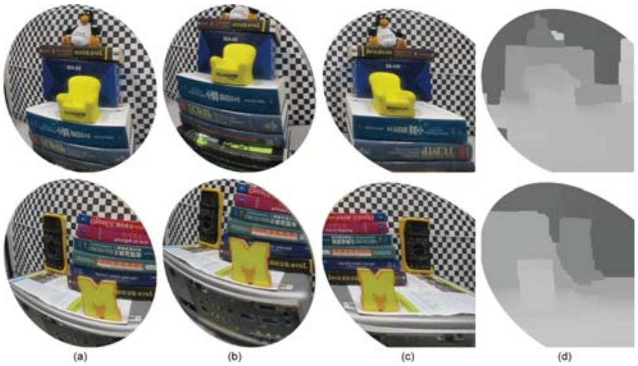

Figure 8. Recovering Real Scenes. Column (a) and (b) show two of the seven mirror image. Column (c) shows the warped virtual perspective image. Column (d) shows the recovered scene geometry using the depth maps.

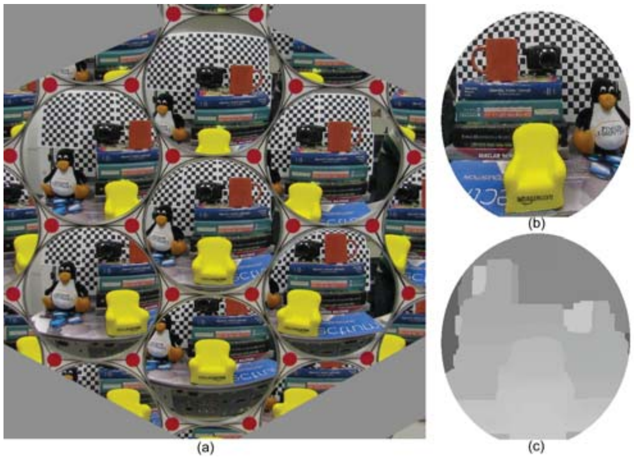

Figure 9. We orient the viewing camera nearly perpendicular to the array. (a) shows captured mirror images. Notice the self-reflection of the camera. (b) warps the central mirror image to a plane to reduce distortions. (c) show the recovered depth map using our volumetric reconstruction.

also developed a multi-perspective volumetric reconstruction algorithm to more accurately recover the 3D scene. Our method constructs the frustum volume with respect to a virtual pinhole camera by first discretizing the scene geometry into voxels and then applying GLC back-projections and graph-cut for carving the voxels. We have demonstrated our framework on both synthetic and real scenes. Experimental results have shown that our methods are reliable and robust.

One limitation of epsilon stereo matching is that the pseudo disparity map cannot be directly translated to a depth map, because the corresponding rays are not guaranteed to intersect in 3D space. In the future, it would be useful to develop new distance metrics to better interpret the resulting two-dimensional disparity map. This problem also draws analogy to the triangulation method [12] for pinhole cameras, where Gaussian noise was used as the usual and reasonable model to achieve projective invariance. In the future, we plan to investigate what noise models should be used to characterize the multi-perspective espilon stereo model.

Our multi-perspective space carving algorithm may also lead to new designs of catadioptric mirrors. For example, one can stitch multiple curved mirrors and capture their reflections of the scene in a single image. Our multi-perspective volumetric reconstruction algorithm can be used to recover dense volumetric 3D scene geometry. The main advantage of our algorithm over existing omni-directional reconstruction methods is that we do not require delicate calibration between the mirror and the camera to maintain single viewpoint ray geometry. Instead, we can decompose the mirror into piecewise GLCs and apply our algorithm on individual GLCs. Our framework may also lead to new catadioptric calibration algorithms based on the GLC decomposition.

Acknowledgement

Ding and Yu were supported in part by NSF grants MSPA-MCS-0625931 and IIS-CAREER-0845268.

References

[1] S. Baker and S. K. Nayar. A theory of single-viewpoint catadioptric image formation. *Int. J. Comput. Vision*, 35(2):175–196, 1999. 2

[2] R. Bhotika, D. J. Fleet, and K. N. Kutulakos. A probabilistic theory of occupancy and emptiness. In *ECCV '02-Part III*, pages 112–132, 2002. 2

[3] Y. Boykov and V. Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. In *ICCV '03*, page 26, 2003. 1

[4] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11):1222–1239, 2001. 1

[5] A. Broadhurst, T. Drummond, and R. Cipolla. A probabilistic framework for space carving. In *ICCV*, pages 388–393, 2001. 2

[6] Y. Ding and J. Yu. Epsilon stereo pairs. In *British Machine Vision Conference I*, pages 162–171, 2007. 1, 4

[7] O. Faugeras, R. Keriven, and K. V. Principles. Variational principles, surface evolution, pdes, level set methods and the stereo problem. *IEEE Transactions on Image Processing*, 7:336–344, 1998. 1

[8] D. Feldman, T. Pajdla, and D. Weinshall. On the epipolar geometry of the crossed-slits projection. In *ICCV '03*, pages 988–995, 2003. 2, 3, 6

[9] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. *Int. J. Comput. Vision*, 70(1):41–54, 2006. 1

[10] R. Gupta and R. I. Hartley. Linear pushbroom cameras. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(9):963–975, 1997. 2

[11] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. 2000. 2

[12] R. I. Hartley and P. Sturm. Triangulation. *Comput. Vis. Image Underst.*, 68(2):146–157, 1997. 7

[13] X. Hou, L.-Y. Wei, H.-Y. Shum, and B. Guo. Real-time multi-perspective rendering on graphics hardware. In *ACM SIGGRAPH 2006 Sketches*, page 79, 2006. 5

[14] S. B. Kang. Catadioptric self-calibration. In *CVPR '00*, 2000. 2

[15] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions using graph cuts. In *ICCV*, pages 508–515, 2001. 1, 4

[16] V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. In *ECCV '02-Part III*, pages 82–96, 2002. 1

[17] K. N. Kutulakos. Approximate n-view stereo. In *ECCV '00-Part I*, pages 67–83, 2000. 2

[18] K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. *Int. J. Comput. Vision*, 38(3):199–218, 2000. 1, 2, 4

[19] D. Lanman, D. E. Crispell, M. Wachs, and G. Taubin. Spherical catadioptric arrays: Construction, multi-view geometry, and calibration. In *3DPVT*, pages 81–88, 2006. 6

[20] A. Motilal and L. Davis. A probabilistic framework for surface reconstruction from multiple images, 2001. 2

[21] S. K. Nayar. Catadioptric omnidirectional camera. In *CVPR '97*, page 482, 1997. 2

[22] T. Pajdla. Stereo with oblique cameras. *Int. J. Comput. Vision*, 47(1-3):161–170, 2002. 1, 2

[23] S. Paris, F. X. Sillion, and L. Quan. A surface reconstruction method using global graph cut optimization. *Int. J. Comput. Vision*, 66(2):141–161, 2006. 1, 6

[24] S. Peleg, M. Ben-Ezra, and Y. Pritch. Omnistereo: Panoramic stereo imaging. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(3):279–290, 2001. 2

[25] M. Pollefeys, R. Koch, and L. V. Gool. A simple and efficient rectification method for general motion. In *ICCV'99-Volume 1*, page 496, 1999. 2

[26] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Comput. Vision*, 47(1-3):7–42, 2002. 2

[27] S. M. Seitz and C. R. Dyer. Photorealistic scene reconstruction by voxel coloring. *Int. J. Comput. Vision*, 35(2):151–173, 1999. 2

[28] S. M. Seitz and J. Kim. The space of all stereo images. *Int. J. Comput. Vision*, 48(1):21–38, 2002. 1, 2, 3

[29] G. G. Slabaugh, R. W. Schafer, and M. C. Hans. Multi-resolution space carving using level sets methods. In *3DPVT*, pages 704–708, 2002. 1

[30] J. Sun, N.-N. Zheng, and H.-Y. Shum. Stereo matching using belief propagation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(7):787–800, 2003. 1

[31] R. Swaminathan, M. D. Grossberg, and S. K. Nayar. Caustics of catadioptric cameras. In *ICCV*, pages 2–9, 2001. 2

[32] J. Yu and L. McMillan. General linear cameras. In *ECCV (2)*, pages 14–27, 2004. 1, 3

[33] J. Yu and L. McMillan. Multiperspective projection and collineation. In *ICCV '05 – Volume 1*, pages 580–587, 2005. 3, 5

[34] A. Zomet, D. Feldman, S. Peleg, and D. Weinshall. Mosaicing new views: The crossed-slits projection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25:741–754, 2003. 2