A Rotational Stereo Model Based on XSlit Imaging

Jinwei Ye Yu Ji Jingyi Yu
University of Delaware, Newark, DE 19716, USA
{jye,yuji,yu}@cis.udel.edu

Abstract

Traditional stereo matching assumes perspective viewing cameras under a translational motion: the second camera is translated away from the first one to create parallax. In this paper, we investigate a different, rotational stereo model on a special multi-perspective camera, the XSlit camera [9, 24]. We show that rotational XSlit (R-XSlit) stereo can be effectively created by fixing the sensor and slit locations but switching the two slits' directions. We first derive the epipolar geometry of R-XSlit in the 4D light field ray space. Our derivation leads to a simple but effective scheme for locating corresponding epipolar "curves". To conduct stereo matching, we further derive a new disparity term in our model and develop a patch-based graph-cut solution. To validate our theory, we assemble an XSlit lens by using a pair of cylindrical lenses coupled with slit-shaped apertures. The XSlit lens can be mounted on commodity cameras where the slit directions are adjustable to form desirable R-XSlit pairs. We show through experiments that R-XSlit provides a potentially advantageous imaging system for conducting fixed-location, dynamic baseline stereo.

1. Introduction

Stereo matching is an extensively studied problem in computer vision [6, 15]. It aims to extract 3D information by examining the relative position from two viewpoints, analogous to the biological stereopsis process. Traditional approaches assume perspective viewing cameras under a translational motion: the second camera is translated away from the first one to have sufficient camera baseline for producing parallax [6]. Input images can be further rectified by projecting onto a common image plane to have purely horizontal parallax [13]. The survey by Scharstein and Szeliski [15] discusses a comprehensive class of state-of-the-art solutions.

In this paper, we investigate a different, rotational stereo model. Instead of translating the camera, we aim to create stereo pairs by rotating the camera, or more precisely, rays collected by the camera. However, rotating a pinhole camera around its center of projection (CoP) results in the same set of rays and does not produce stereo pairs. We therefore focus on creating rotational stereo using non-pinhole or multi-perspective cameras [23].

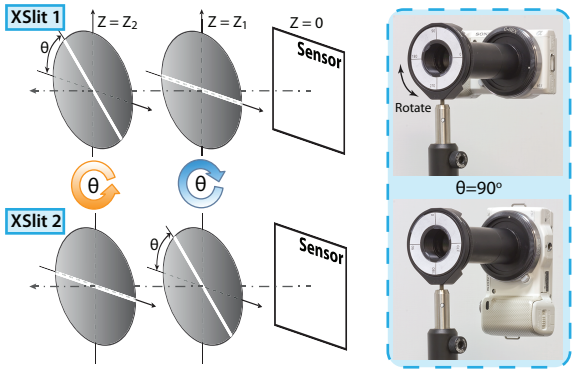

Figure 1. Left: an illustration of the rotational XSlit stereo model. Right: our physical implementation using an XSlit lens.

A multi-perspective camera captures rays originating from different points in space [18, 23]. Such imaging models widely exist in nature, *e.g.*, a compound insect eye can consist of thousands of individual photoreceptor units pointing in slightly different directions. The collected rays by these "cameras" generally do not pass through a common CoP and hence do not follow pinhole geometry. Unlike the pinhole case, a multi-perspective camera can be rotated to acquire a different set of rays. When properly configured, the resulting ray geometry is potentially amenable for stereo matching.

There have been significant advances on the theory of multi-perspective stereo in the past decade. Seitz [16, 17] characterized all possible multi-perspective stereo pairs and concluded the epipolar geometry, if it exists, has to be a doubly ruled surface. Therefore, only a small variety of multi-perspective stereo pairs exist. Pajdla [10, 11, 12] independently obtained the same results and further studied stereo matching on the multi-perspective linear oblique camera. Their results show that a small variety of multi-perspective stereo pairs exist. In this paper, we present a practical multi-perspective stereo solution based on a special class of multi-perspective cameras, the XSlit camera [9, 24].

An XSlit camera collects rays simultaneously passing through two oblique lines (slits) in 3D space. Feldman *et al*. [4] derived the translational XSlit stereo model: an XSlit camera can be translated along one of the two slits to form valid stereo pairs with purely horizontal parallax. In this paper, we show that, instead of translating the XSlit cameras, we can form valid stereo pairs by fixing the sensor/slit locations but switching the slits' directions. We call this model rotational XSlit stereo or R-XSlit stereo. We first present a theoretical analysis to characterize R-XSlit epipolar geometry. While previous analysis was carried out in 3D geometry space [4, 10, 12, 11, 17], ours is derived in the 4D light field ray space [8, 22]. Our derivation also leads to simple but effective schemes for locating corresponding epipolar "curves" and analyzing recoverable depth range and depth error. For stereo matching, we further derive a new R-XSlit disparity term and develop a patch-based graph-cut solution.

We validate our theory and algorithms on synthetic and real data. For real scenes, we assemble an XSlit lens using a pair of cylindrical lenses coupled with slit-shaped apertures. The XSlit lens can be mounted on commodity cameras where the slit direction can be changed to form an R-XSlit pair. We show through experiments that R-XSlit provides a potentially advantageous stereo imaging system. In particularly, it can achieve "fixed-location" stereo by rotating only the slits, hence eliminating the need of placing two cameras at different spatial locations in perspective stereo.

2. R-XSlit Stereo Model

An XSlit camera collects rays that simultaneously pass through two oblique (neither parallel nor coplanar) slits in 3D space [9, 24]. The ray geometry of XSlit has been previous studied using XSlit projection matrix [24], linear oblique [9], light field parametrization [22], or ray regulus [14]. In this paper, we adopt the light field two-plane parametrization [8, 22] for its simplicity. Specifically, we choose two planes Π_{uv} and Π_{st} parallel to both slits but containing neither slits. Next, we orthogonally project both slits on Π_{uv} and use their intersection point as the origin of the coordinate system.

To further simplify our analysis, we use the $[u, v, \sigma, \tau]$ parametrization where $\sigma = s - u$ and $\tau = t - v$. We choose Π_{uv} as the default image (sensor) plane so that (u, v) can be directly used as the pixel coordinate and $(\sigma, \tau, 1)$ can be viewed as the direction of the ray. We assume that the two slits, l_1 and l_2, lie at $z = Z_1$ and $z = Z_2$ and have angle θ_1 and θ_2 w.r.t. the x-axis, where $Z_2 > Z_1 > 0$ and $\theta_1 \neq \theta_2$. Therefore, each XSlit camera can be represented as $\mathcal{C}(Z_1, Z_2, \theta_1, \theta_2)$. Each pixel (u, v) in $\mathcal{C}$ maps to a ray with direction $(\sigma, \tau, 1)$ (see Appendix A) as

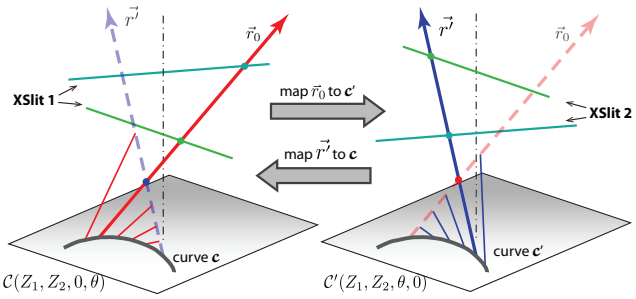

Figure 2. Epipolar curves and geometry in an R-XSlit stereo pair.

$$\begin{cases} \sigma = (Au + Bv)/E \\ \tau = (Cu + Dv)/E \end{cases} \quad (1)$$

where

$$A = Z_2 \cos\theta_2 \sin\theta_1 - Z_1 \cos\theta_1 \sin\theta_2, \quad B = (Z_1 - Z_2) \cos\theta_1 \cos\theta_2,$$
$$C = (Z_1 - Z_2) \sin\theta_1 \sin\theta_2, \quad D = Z_1 \cos\theta_2 \sin\theta_1 - Z_2 \cos\theta_1 \sin\theta_2,$$
$$E = Z_1 Z_2 \sin(\theta_2 - \theta_1)$$

A rotational XSlit or R-XSlit pair consists of two XSlit cameras, XSlit 1: $\mathcal{C}(Z_1, Z_2, \theta_1, \theta_2)$ and XSlit 2: $\mathcal{C}'(Z_1, Z_2, \theta_2, \theta_1)$, *i.e*., the two slits switch their directions as shown in Fig. 1. We can further simplify this model by rotating the coordinate system to align l_1 in $\mathcal{C}$ (or l_2' in $\mathcal{C}'$) with the x-axis. The R-XSlit pair is then simplified as XSlit 1: $\mathcal{C}(Z_1, Z_2, 0, \theta)$ and XSlit 2: $\mathcal{C}'(Z_1, Z_2, \theta, 0)$, where $\theta = \theta_2 - \theta_1$. We use $\mathcal{P}(Z_1, Z_2, \theta)$ to represent an R-XSlit pair.

3. R-XSlit Stereo Matching

Next we derive the epipolar geometry in an R-XSlit pair. Although the general theory behind multi-perspective stereo is well known [4, 10, 11, 12, 17], *i.e*., only three varieties of epipolar geometry exist: planes, hyperboloids, and hyperbolic-paraboloids, effectively testing whether a pair of multi-perspective cameras form valid epipoplar geometry is still a challenging problem. Our approach is to first locate potential epipolar curves on corresponding images and then determine if the two curves form valid epipolar geometry.

3.1. Existence

To find potential epipolar curves in an R-XSlit pair $\mathcal{P}(Z_1, Z_2, \theta)$, we first trace out a ray $\vec{r}_0[u_0, v_0, \sigma_0, \tau_0]$ from $\mathcal{C}(Z_1, Z_2, 0, \theta)$ in $\mathcal{P}$. If epipolar geometry exists, there should exist a curve in $\mathcal{C}'(Z_1, Z_2, \theta, 0)$ where all rays originating from the curve intersect with $\vec{r}_0$. This implies that we can directly project $\vec{r}_0$ into $\mathcal{C}'$ by using the XSlit line projection equation (see Appendix C) as curve c':

$$\sin\theta \cdot u'v' - \cos\theta \cdot v'^2 = \sin\theta \cdot u_0 v_0 - \cos\theta \cdot v_0^2 \quad (2)$$

Similarly, we can pick an arbitrary ray $\vec{r}'$ originated from c' and project it back to $\mathcal{C}$, as shown in Fig. 2. The resulting curve c in $\mathcal{C}$ is then

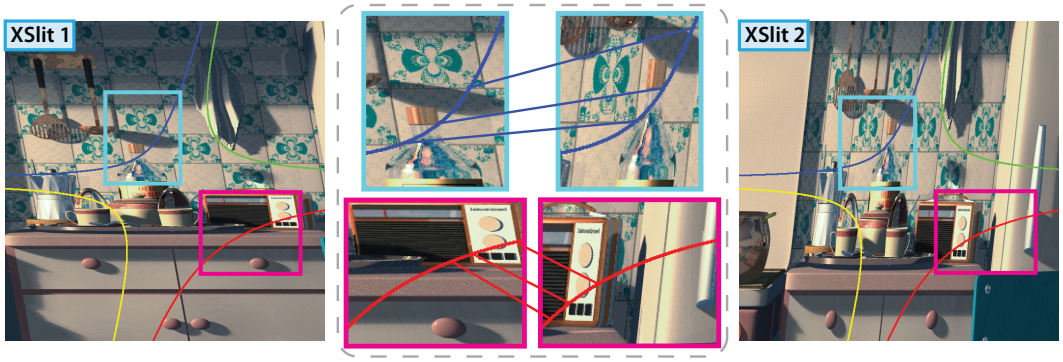

Figure 3. Correspondence matching. Four pairs of epipolar curves (hyperbolas) are plotted on an R-XSlit stereo pair of a kitchen scene. The close-up views (middle) show the corresponding feature points.

$$\sin\theta \cdot uv - \cos\theta \cdot v^2 = \sin\theta \cdot u_0 v_0 - \cos\theta \cdot v_0^2 \quad (3)$$

To determine if these rays form valid epipolar geometry, we carry out a ray geometry analysis. Specifically, we first derive the ray-ray intersection criteria. Recall that if two rays $[u_1, v_1, \sigma_1, \tau_1]$ and $[u_2, v_2, \sigma_2, \tau_2]$ intersect, there must exist some λ_1 and λ_2 so that

$$[u_1, v_1, 0] + \lambda_1[\sigma_1, \tau_1, 1] = [u_2, v_2, 0] + \lambda_2[\sigma_2, \tau_2, 1]$$

Eliminating λ_1 and λ_2, we have the ray-ray intersection criteria:

$$\frac{u_1 - u_2}{v_1 - v_2} = \frac{\sigma_1 - \sigma_2}{\tau_1 - \tau_2} \quad (4)$$

Theorem 1. *The epipolar curves in an R-XSlit pair* $\mathcal{P}(Z_1, Z_2, \theta)$ *are* $\sin\theta \cdot uv - \cos\theta \cdot v^2 = \kappa$ *in both XSlit cameras, where* κ *is some constant.*

Proof. We prove that every pair of rays $\vec{r}[u, v, \sigma, \tau]$ from c and $\vec{r'}[u', v', \sigma', \tau']$ from c' satisfy the ray-ray intersection criteria. We first rewrite Eqn. (2) and (3) as

$$u = \frac{\cos\theta \cdot v}{\sin\theta} + \frac{\kappa}{\sin\theta \cdot v} \quad (5)$$

By substituting u and u' with v and v', the LHS of Eqn. (4) becomes

$$\frac{u - u'}{v - v'} = \frac{\cos\theta}{\sin\theta} - \frac{\kappa}{\sin\theta \cdot vv'}$$

To compute the RHS Eqn. (4), we use the ray constraints in XSlit camera (Eqn. (1)) and we have

$$\frac{\sigma - \sigma'}{\tau - \tau'} = \frac{(Au + Bv) - (A'u' + B'v')}{(Cu + Dv) - (C'u' + D'v')} = \frac{\cos\theta}{\sin\theta} - \frac{\kappa}{\sin\theta \cdot vv'}$$

Therefore, we have $\frac{u - u'}{v - v'} = \frac{\sigma - \sigma'}{\tau - \tau'}$, *i.e.*, $\vec{r}$ and $\vec{r'}$ satisfy the ray-ray intersection constraint. □

Theorem 1 reveals that, different from the perspective stereo, the epipolar "lines" in our R-XSlit pair are hyperbolas. The search space of correspondences, however, is still effectively reduced to 1D. Fig. 3 shows several epipolar curves in an R-XSlit pair. Notice that, although our analysis focuses on R-XSlit stereo, it can also be used to prove the translational XSlit stereo condition [4]. R-XSlit can also be viewed as a special case of the second XSlit condition in [4] where the four slits intersect at four distinct points. In an R-XSlit pair, slits l_1 and l_2' (also l_1' and l_2) intersect at infinity.

3.2. Disparity

Next, we develop R-XSlit stereo matching algorithm. In traditional perspective stereo, disparity is defined as purely horizontal parallax. However, in our R-XSlit pair, corresponding pixels exhibit both vertical parallax and horizontal parallax as the epipolar curves are hyperbolas. We therefore need to redefine disparity.

Recall that valid disparity definition should satisfy three criterion: 1) the disparity should only depend on object depth; 2) it should be a monotonic function in object depth; and 3) it can be used to locate the corresponding pixel in the second view. Let us first study the images of a scene point in an R-XSlit pair. Given a 3D point $\mathbf{X} = (x, y, z)$, we can compute its images in $\mathcal{P}(Z_1, Z_2, \theta)$, *i.e.*, $p = (u, v)$ in $\mathcal{C}$ and $p' = (u', v')$ in $\mathcal{C}'$, using the XSlit point projection equation (see Appendix B) as:

$$u = \frac{Z_2 x}{Z_2 - z} - \frac{\cos\theta}{\sin\theta} \cdot \frac{(Z_1 - Z_2)yz}{(Z_1 - z)(Z_2 - z)}, \; v = \frac{Z_1 y}{(Z_1 - z)} \quad (6)$$

and

$$u' = \frac{Z_1 x}{Z_1 - z} + \frac{\cos\theta}{\sin\theta} \cdot \frac{(Z_1 - Z_2)yz}{(Z_1 - z)(Z_2 - z)}, \; v' = \frac{Z_2 y}{(Z_2 - z)} \quad (7)$$

To satisfy criteria 1), the disparity should not contain x and y terms. We therefore define the XSlit disparity as:

$$d^{\text{XS}} = \frac{v'}{v} = \frac{Z_2}{Z_1} \cdot \frac{z - Z_1}{z - Z_2} \quad (8)$$

It is easy to see that d^{XS} is monotonically decreasing in z for $z > Z_2$ and therefore satisfy disparity criteria 2). Finally, to enable correspondence matching, given a pixel (u_p, v_p) in $\mathcal{C}$ and its disparity d_p^{XS} w.r.t. $\mathcal{C}'$, we can reuse the epipolar curve constraint (Eqn. (5)) to find its corresponding pixel (u'_p, v'_p) in $\mathcal{C}'$. Specifically, we can compute $v'_p = v_p \cdot d_p^{\text{XS}}$ and then apply the epipolar curve constraint (Eqn. (5)) to compute $u'_p = (\cos\theta \cdot v'_p)/\sin\theta + \kappa/(\sin\theta \cdot v'_p)$, where $\kappa = \sin\theta \cdot u_p v_p - \cos\theta \cdot v_p^2$.

In perspective cameras, the singularity of disparity occurs when scene points lie on the line connecting the two CoPs, *i.e.*, rays from the two cameras become identical. From Eqn. (8), we observe that an R-XSlit pair has singularity at $v = 0$ where disparity can no longer be computed. In reality, $v = 0$ implies $y = 0$ as shown in Eqn. (6) and (7), *i.e.*, epipolar geometry still exists and it corresponds to the $y = 0$ plane. In that case, we can redefine the disparity as $d^{\text{XS}} = u/u'$, which is consistent with v'/v when $y = 0$. The real singularity is when $x = y = 0$, *i.e.*, the ray aligns with the z-axis which is the only ray shared by both XSlit cameras.

3.3. Graph-Cut Stereo Matching

To recover depth from our R-XSlit pair, we reuse the graph-cut algorithm [1, 2, 7] by modeling stereo matching as XSlit disparity labeling. Specifically, we discretize the disparity d^{XS} (Eqn. (8)) to M labels. Given a label $d_i^{\text{XS}}, i \in [1, M]$ to a pixel p in $\mathcal{C}$, we can find its corresponding pixel $p' = d_i^{\text{XS}}(p)$ in $\mathcal{C}'$ as described in Section 3.2. The energy function E of assigning a label d_i^{XS} to a pixel p in $\mathcal{C}$ is identical to the one used in perspective stereo matching:

$$E(d_i^{\text{XS}}) = \alpha \cdot \sum_{p \in P} E_d(p, d_i^{\text{XS}}(p)) + \sum_{p_1, p_2 \in N} E_s(p_1(d_i^{\text{XS}}), p_2(d_j^{\text{XS}}))$$

where P is the set of all pixels in $\mathcal{C}$, N represents the pixel neighborhood, and the non-negative coefficient α balances the data term $E_d(p) = \|I(p) - I'(d_i^{\text{XS}}(p))\|$ and the smooth term E_s.

Once we recover the disparity map, we can compute the object depth z by inverting Eqn. (8) as

$$z = Z_2\left(1 + \frac{Z_2 - Z_1}{Z_1 d^{\text{XS}} - Z_2}\right) \tag{9}$$

Notice that Eqn. (9) applies to pixels both on and off the v-axis.

The pixel-wise comparison of the data term can be sensitive to camera alignment and image noise. It is common to compare patch similarity to improve robustness. Different from perspective stereo, image patches in an XSlit image are distorted (sheared and stretched), where the distortion is determined by the slit position/direction and object depth [3, 24]. We therefore first correct such distortions and then measure patch similarity.

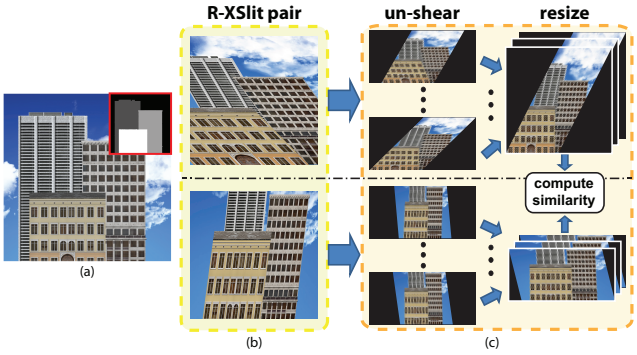

Figure 4. Distortion correction in patch-based stereo matching. (a) shows a perspective view of the scene and its depth map; (b) shows an R-XSlit stereo pair; (c) for robust patch matching, we first "un-shear" the two images given a specific depth label and then resize them to compute similarity.

Our distortion correction procedure consists of two steps: we first "un-shear" the patches and then resize them to have the same aspect ratio. Specifically, when assigning a disparity label d_i^{XS} to a pixel in camera $\mathcal{C}$, we first shear the patches in each XSlit view with a shear matrix $\left(\begin{smallmatrix} 1 & 0 \\ s & 1 \end{smallmatrix}\right)$, where s is the shear factor. For $\mathcal{C}$, $s = \frac{\cos\theta}{\sin\theta} \cdot \frac{z_i(Z_1 - Z_2)}{Z_1(z_i - Z_2)}$; and for $\mathcal{C}'$, $s' = \frac{\cos\theta}{\sin\theta} \cdot \frac{z_i(Z_2 - Z_1)}{Z_2(z_i - Z_1)}$, where z_i is the scene depth corresponding to d_i^{XS}.

Next, we correct aspect ratio distortion. Recall that for a scene point at depth z_i, its aspect ratio in $\mathcal{C}$ can be computed as $\frac{Z_2(z_i - Z_1)}{Z_1(z_i - Z_2)}$ and in $\mathcal{C}'$ as $\frac{Z_1(z_i - Z_2)}{Z_2(z_i - Z_1)}$. By Eqn. (8), the aspect ratio is identical to the disparity d_i^{XS} corresponding to z_i. Therefore we can directly use d_i^{XS} as the scaling factor. Assume the original image resolutions are $m \times n$ in $\mathcal{C}$ and $n \times m$ in $\mathcal{C}'$, we first resize the first to $d_i^{\text{XS}} m \times n$ and second to $n \times d_i^{\text{XS}} m$. We then query the patches of the same size from the resized results for computing the data term. For acceleration, we further pre-scale the input image pairs with different disparity labels and then fetch patches from the corresponding ones given a specific disparity label. The complete distortion correction process is shown in Fig. 4.

Fig. 5 shows a sample stereo matching result using our approach on an R-XSlit pair $\mathcal{P}(1.0, 1.5, 105°)$. The images are synthesized using the POV-Ray ray tracer (www.povray.org) with a general XSlit camera model. The scene has depth range of $[6, 35]$. We also add Gaussian noise of $\sigma = 0.05$ to the rendered XSlit images. Fig. 5(c) shows the pixel-based result using graph-cut. Fig. 5(d) and (e) show the patch-based results with and without distortion correction. We observe that pixel-based result lacks smoothness with image noise while patch-based result without distortion correction produces large errors.

Figure 5. Stereo matching on an R-XSlit pair. (a) and (b) are the input XSlit images with the ground truth disparity map shown at the left-top corner of (a); (c)-(e) are the recovered disparity maps using pixel-based (c), patch-based with distortion correction (d), and patch-based without distortion correction (e) schemes.

4. Axis-Aligned R-XSlit Stereo

A special R-XSlit stereo model is when the two slits are orthogonal and axis-aligned. This is commonly referred to as the parallel orthogonal XSlit (POXSlit) camera [24]. It corresponds to an R-XSlit pair with $\theta = 90°$ and we call it an R-POXSlit pair. By Theorem 1, we obtain the epipolar curves as: $uv = \kappa$. As shown in the following sections, the R-POXSlit stereo pair has a number of advantages. First, POXSlit cameras can be physically constructed using special lenses (Section 4.1). Second, images of a POXSlit camera appear similar to perspective ones with fewer distortions. Finally, instead of rotating the two slits individually, we can rotate the camera by $90°$ to form an R-POXSlit pair.

4.1. Camera Construction

The idea of constructing real XSlit cameras can be backdated to the 18th century. The crossed-slit anamorphoser, credited to Ducos du Hauron, modifies pinhole camera by replacing the pinhole with a pair of narrow, perpendicularly crossed slits, spaced apart along the camera axis [19]. Image distortions appear anamorphic or anamorphotic and the degree of anamorphic compression closely matches the estimated distortion using the crossed-slit model. Similar to lensless pinhole cameras, this brute-force implementation of XSlit suffers from low light efficiency and poor imaging quality.

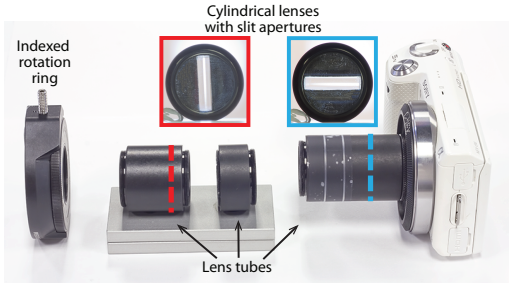

Figure 6. Our physical implementation of the R-POXSlit pair.

Today a commodity camera uses spherical thin lens to emulate a pinhole camera by focusing rays passing through the lens on to a 3D point. Similarly, we design a special XSlit lens. We observe that a cylindrical lens is a section of a cylinder that focuses rays passing through it onto a line parallel to the intersection of the surface of the lens and a plane tangent to it. The lens compresses the image in the direction perpendicular to this line, and leaves it unaltered in the direction parallel to it (in the tangent plane). This implies that we can concatenate two layers of cylindrical lenses to synthesize an XSlit lens. To further increase the XSlit camera's depth-of-field, we couple the lens with slit-shaped apertures.

Fig. 6 illustrates our prototype POXSlit camera where we mount the XSlit lens on a commodity interchangable lens camera (*e.g.*, Sony NEX-5N). We align the two cylindrical lenses orthogonally using a lens tube. To produce an R-POXSlit pair, the brute-force approach is to rotate each individual lens. This, however, poses challenges on accurate alignment. We, instead, mount the camera onto an indexed rotation ring and capture the scene twice by rotating the camera by $90°$.

4.2. Depth Range and Error

To evaluate the practicability of our R-POXSlit stereo, an important task is to measure the depth range and error in comparison with perspective stereo [5, 20]. In our analysis, we assume that both stereo models has the same (1D) pixel size ϵ_p.

In perspective stereo, we assume the two cameras have identical focal length Z_f and are separated by baseline b. The object depth z and its disparity d are correlated by $z = Z_f(1 + b/d)$. The maximum recoverable depth and depth error (without considering sub-pixel accuracy) are $Z_f(1 + b/\epsilon_p)$ and $(z - Z_f)^2\epsilon_p/(bZ_f)$ respectively.

In an R-POXSlit pair $\mathcal{P}(Z_1, Z_2, 90°)$, we assume scene depth $z > Z_2$ so that $d^{\text{XS}} > 0$ as shown in Eqn. (8). To study the maximum recoverable depth and depth error in R-POXSlit stereo, we conduct a *pixel-shift analysis*. We first consider the disparity change Δd^{XS} by shifting one pixel along the epipolar curve. Given a pixel (u, v) in $\mathcal{C}$ and its disparity d^{XS}, we can calculate its correspondence in $\mathcal{C}'$ as

$(u', v') = (u/d^{\text{XS}}, v \cdot d^{\text{XS}})$. Our goal is to test if (u', v') shifts by one pixel, how much disparity (depth) changes would occur. Without loss of generality, if we shift v' by one pixel, we can locate a new pixel on the epipolar curve as $(\tilde{u}', \tilde{v}') = (\kappa/(v' + \epsilon_p), v' + \epsilon_p)$. We can then compute the corresponding disparity $\tilde{d}^{\text{XS}} = \tilde{v}'/v = (v' + \epsilon_p)/v$. Therefore, we have $\Delta d^{\text{XS}} = \tilde{d}^{\text{XS}} - d^{\text{XS}} = \epsilon_p/v$. By Eqn. (9), the depth error can then be computed as

$$\begin{aligned}\Delta z &= Z_2(1 + \frac{Z_2 - Z_1}{Z_1 d^{\text{XS}} - Z_2}) - Z_2(1 + \frac{Z_2 - Z_1}{Z_1(d^{\text{XS}} + \Delta d^{\text{XS}}) - Z_2}) \\ &\approx \frac{Z_1(z - Z_2)^2}{Z_2(Z_2 - Z_1)} \cdot \frac{\epsilon_p}{v}\end{aligned} \tag{10}$$

Eqn. (10) illustrates that the depth error in R-POXSlit stereo is similar to the one in perspective case in that it is linear in ϵ_p and quadratic in z. However, in perspective stereo, its minimum disparity change is identical (*i.e.*, ϵ_p) across all pixels whereas in R-POXSlit it is pixel-dependent (*i.e.*, ϵ_p/v). This can be interpreted in terms of the epipolar geometry. In perspective stereo, the epipolar geometry is a plane on which rays form a perspective uniform lattice. In contrast, the epipolar geometry in R-POXSlit stereo is a hyperboloid where the depth variation under uniform v sampling is non-linear.

We can further compute the maximum recoverable depth z_{max}. To do so, we first compute the disparity and correspondence (u'_∞, v'_∞) for $z \to \infty$ and then shift one pixel from (u'_∞, v'_∞) along the epipolar curve. Specifically by Eqn. (8), we have the infinity disparity $d^{\text{XS}}_\infty = Z_2/Z_1$ when $z \to \infty$ (notice that this is different from the perspective case that $d^{\text{XS}}_\infty = 0$). We then reuse Eqn. (9) to compute z_{max} as

$$\begin{aligned}z_{max} &= Z_2(1 + \frac{Z_2 - Z_1}{Z_1(d^{\text{XS}}_\infty + \Delta d^{\text{XS}}) - Z_2}) \\ &= Z_2(1 + \frac{Z_2 - Z_1}{Z_1} \cdot \frac{v}{\epsilon_p})\end{aligned} \tag{11}$$

The maximum depths for R-POXSlit and perspective are both inverse proportional to ϵ_p. However, same as the depth error, it also varies with respect to v, *i.e.*, the farther away the pixel from the v-axis, the larger the resolvable depth.

Based on our analysis, we can define a virtual baseline in R-POXSlit camera as $b^{\text{XS}} = Z_2/Z_1$. We can then rewrite Eqn. (10) and Eqn. (11) w.r.t. b^{XS} as

$$\begin{aligned}\Delta z &= \frac{(z - Z_2)^2}{Z_2(b^{\text{XS}} - 1)} \cdot \frac{\epsilon_p}{v} \\ z_{max} &= Z_2(1 + (b^{\text{XS}} - 1) \cdot \frac{v}{\epsilon_p})\end{aligned} \tag{12}$$

Eqn. (12) also reveals that, same as perspective stereo, the larger the XSlit base line, the larger the maximal resolvable depth z_{max} and the smaller the depth error Δz. While perspective stereo needs to physically separate the cameras apart for increasing the baseline, R-POXSlit stereo can fix the sensor location but separates the two slits further away. This implies that we can potentially conduct fixed-location, dynamic baseline stereo.

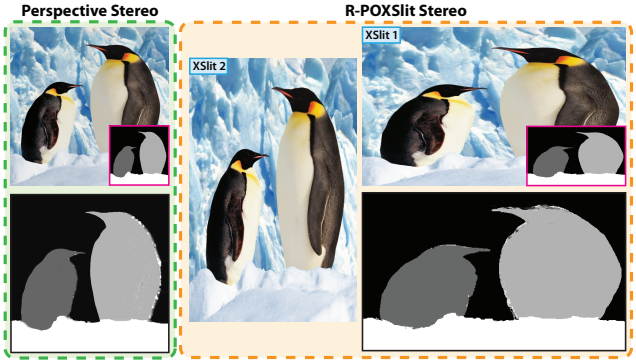

Figure 7. Perspective (left) vs. R-POXSlit (right) stereo matching results on a synthetic scene.

4.3. Experiments

We have validated our R-POXSlit stereo on both synthetic and real data.

Synthetic Data. We first test our algorithm on synthetic data rendered by the POV-Ray ray tracer. We have extended the camera model in POV-Ray by implementing a general XSlit camera model. Fig. 7 shows a rendered R-POXSlit pair $\mathcal{P}(1.0, 1.5, 90°)$ that captures a scene composed of four depth layers of $[3, 16]$. The images were rendered at a resolution of 600×380. We discretize the XSlit disparity to ten labels from 1.55 to 2.0 with step 0.05 and apply the distortion-corrected patch-based graph-cut algorithm described in Section 3.3 to recover the scene depth.

In this example, we do not conduct shear correction step since there is little shearing distortion for frontal-parallel objects in an POXSlit. We still conduct aspect-ratio correction (Section 3.3) and then apply our patch-based stereo matching with patch size 5×5. We have further compared R-XSlit stereo with traditional perspective stereo where we assume that the CoP is at 1.5 (*i.e.*, the location of Z_2) and the camera baseline is 0.5 (*i.e.*, the distance between the two slits). Fig. 7 shows our R-POXSlit recovered disparity map which is comparable to the perspective stereo result.

Real Data. Next, we validate our approach on scenes acquired by our prototype POXSlit camera (Section 4.1). For proof-of-concept, we first acquire a simple indoor scene composed of roughly five depth layers. Fig. 8 shows our experimental setup. We capture the scene twice by rotating the camera by 90° on a rotation ring to generate the R-POXSlit pair. The XSlit images are captured at resolution of 2448×1376 and down-sampled to half of its original resolution. The two slits' positions w.r.t. the image sensor are $Z_1 = 38mm$ and $Z_2 = 66mm$ and have width of $2mm$.

Figure 8. An R-POXSlit stereo pair of a study table scene captured by our prototype POXSlit camera.

It is important to note that the rotation of the ring does not guarantee that the optical axis (*i.e.*, the central ray) is perfectly aligned. However, we can still apply our distortion-corrected patch-based graph-cut algorithm to recover a disparity map from the POXSlit pair. This is analogous to conducting stereo matching on perspective image pairs that are slightly misaligned. The misalignment can lead to inaccurate depth maps, although the recovered disparity map still reveals meaningful scene structures.

In this example, we discretize the disparity label into 20 levels at range of $[1.8, 2.3]$ and apply patch-based stereo matching. In Fig. 9(a), we use a relatively small XSlit baseline ($b^{\text{XS}} = 1.7$). As a result, the maximum resolvable depth is relatively small and depth error is relatively large (Section 4.2). For example, it is unable to distinguish the computer graphics book and the patterned background, as shown in Fig. 9(b).

We then increase the XSlit baseline by adjusting Z_2 to $76mm$ with the same Z_1 fixed. The new baseline is now *i.e.*, $b^{\text{XS}} = 2$. By Eqn. (12), we should be able to increase the maximum resolvable depth while reducing depth errors. Fig. 9(d) shows the result with the new baseline. The background and the book are now separately detected as two layers. The new R-POXSlit images, however, have a narrower field-of-view. Further, they exhibit stronger distortions, *e.g.*, Fig. 9(c) is more horizontally stretched than Fig. 9(a).

Finally, we demonstrate our technique on a deep scene composed of complex materials and lighting. The challenge here is the limited depth-of-field. In our experiments, we first approximate the average scene depth for focusing the lens. If scene depth variation is small, the images will

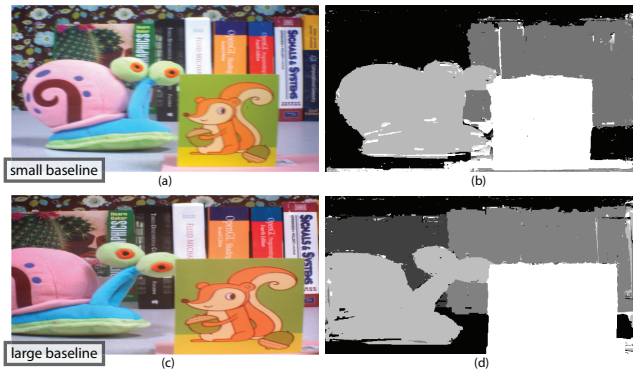

Figure 9. Stereo matching results on the study table scene. (a) and (c) are acquired with XSlit baseline 1.7 and 2 respectively. (b) and (d) show their corresponding disparity maps recovered by our algorithm.

appear all-focused. However, for a deep scene such as Fig. 11, if we use the same slit aperture setting ($2mm$) as in the previous example, the background appears defocused. Moreover, the defocus kernels of the same region appear significantly different in the two XSlit images, one horizontal and the other vertical. We therefore use a narrower aperture of width $1mm$. To guarantee sufficient exposure, the images are captured with longer exposure time ($1/10s$) under ISO 400. The background now appear nearly focused and our stereo reconstruction algorithm produces a reasonable disparity map estimation as shown Fig. 10.

5. Discussions and Future Work

We have presented a new rotational stereo model based on the XSlit camera. This rotational XSlit or R-XSlit pair can be effectively created by fixing the sensor location while strategically rotating the two slits. On the theory front, we have derived the R-XSlit epipolar geometry under the 4D light field. We have shown that the corresponding epipolar "curves" are hyperbolas and we have developed a robust patch-based stereo matching algorithm to handle image distortions. A special R-XSlit pair is when the two slits are orthogonal. We have presented its physical implementations using XSlit lenses and discussed its depth range and error.

There are a number of future directions we plan to explore. First, our prototype R-XSlit pair requires rotating the camera to capture the scene twice. It, therefore, cannot handle dynamic scenes. For slow motion targets, a possible solution is to mount the camera on a fast rotating motor and synchronize the capture and rotation. Second, similar to perspective stereo, non-frontal parallel objects impose challenges in R-XSlit. In particular, shear correction used in our stereo matching algorithm can lead to large errors on slanted planar objects. In the future, we plan to integrate recently proposed XSlit shape-from-distortion technique [21] with stereo matching to robustly handle such scenes.

Figure 10. Stereo matching results of a deep ourdoor scene. Left: one of the XSlit images acquired with slits of width $1mm$. Right: our recovered disparity map.

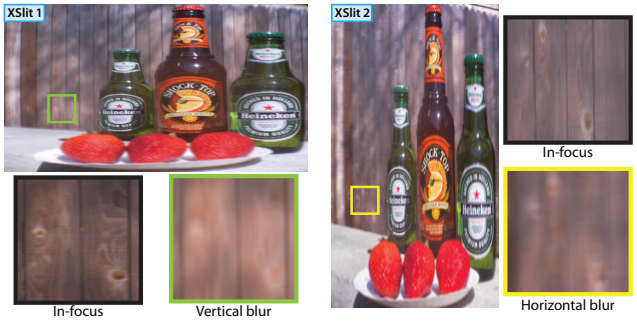

Figure 11. An R-XSlit pair captured with slit apertures of width $2mm$. The images have a shallow depth-of-field, with one XSlit (left) exhibiting horizontal blurs while the second (right) vertical blurs at the background.

As discussed in Section 4.3, defocus blur can be a major artifact in our prototype XSlit camera. A unique characteristics in XSlit defocus blur is that the shape of the blur kernel is depth-dependent and appears differently in the two XSlit images. Our current solution is to use a small aperture. This special phenomenon, however, may lead to new depth-from-defocus solutions, *e.g.*, one can potentially analyze blur variations across the XSlit images to infer depth.

Acknowledgements

We thank Mohit Gupta and Shree Nayar for their invaluable inputs and suggestions. This project was partially supported by the National Science Foundation under grants IIS-CAREER-0845268 and IIS-RI-1016395.

References

[1] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, September 2004.

[2] Y. Boykov, O. Veksler, and R. Zabih. Efficient approximate energy minimization via graph cuts. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 20(12):1222–1239, November 2001.

[3] Y. Ding and J. Yu. Multiperspective distortion correction using collineations. In *Asian Conference on Computer Vision (ACCV)*, 2007.

[4] D. Feldman, T. Pajdla, and D. Weinshall. On the epipolar geometry of the crossed-slits projection. In *IEEE International Conference on Computer Vision (ICCV)*, 2003.

[5] D. Gallup, J.-M. Frahm, P. Mordohai, and M. Pollefeys. Variable baseline/resolution stereo. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[6] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, NY, USA, 2 edition, 2003.

[7] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, February 2004.

[8] M. Levoy and P. Hanrahan. Light field rendering. In *ACM SIGGRAPH*, pages 31–42, 1996.

[9] T. Pajdla. Geometry of two-slit camera. Technical Report CTU-CMP-2002-02, Czech Technical University.

[10] T. Pajdla. Epipolar geometry of some non-classical cameras. In *Proc. of Computer Vision Winter Workshop*, Slovenian Pattern Recognition Society, pages 223–233, 2001.

[11] T. Pajdla. Stereo with oblique cameras. In *IEEE Workshop on Stereo and Multi-Baseline Vision*, pages 85–91, 2001.

[12] T. Pajdla. Stereo with oblique cameras. *International Journal on Computer Vision*, 47(1-3):161–170, April 2002.

[13] M. Pollefeys, R. Koch, and L. J. V. Gool. A simple and efficient rectification method for general motion. In *IEEE International Conference on Computer Vision (ICCV)*, 1999.

[14] J. Ponce. What is a camera? In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[15] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal on Computer Vision*, 47(1-3):7–42, April 2002.

[16] S. M. Seitz. The space of all stereo images. In *IEEE International Conference on Computer Vision (ICCV)*, 2001.

[17] S. M. Seitz and J. Kim. The space of all stereo images. *International Journal on Computer Vision*, 48(1):21–38, June 2002.

[18] S. M. Seitz and J. Kim. Multiperspective imaging. *IEEE Computer Graphics and Applications*, 23(6):16–19, November 2003.

[19] T. Storms. *The Crossed-slit Anamorphoser: An Analysis of Its Characteristics and Utility in Cartography*. University of Washington, 1981.

[20] R. Szeliski and D. Scharstein. Sampling the disparity space image. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 26(3):419–425, March 2004.

[21] J. Ye, Y. Ji, and J. Yu. Manhattan scene understanding via xslit imaging. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[22] J. Yu and L. McMillan. General linear cameras. In *European Conference on Computer Vision (ECCV)*, 2004.

[23] J. Yu, L. McMillan, and P. Sturm. Multi-perspective modelling, rendering and imaging. *Computer Graphics Forum*, 29(1):227–246, 2010.

[24] A. Zomet, D. Feldman, S. Peleg, and D. Weinshall. Mosaicing new views: the crossed-slits projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(6):741–754, June 2003.

Appendix A: XSlit Ray Constraints

Given a XSlit Camera $\mathcal{C}(Z_1, Z_2, \theta_1, \theta_2)$ with two slits, l_1 and l_2, lying at $z = Z_1$ and $z = Z_2$ and having angles θ_1 and θ_2 w.r.t.the x-axis, for each ray $[u, v, \sigma, \tau]$ in $\mathcal{C}(Z_1, Z_2, \theta_1, \theta_2)$, there must exist some λ_1 and λ_2 so that

$$\begin{cases} u + Z_1\sigma = \lambda_1 \cos\theta_1;\ v + Z_1\tau = \lambda_1 \sin\theta_1 \\ u + Z_2\sigma = \lambda_2 \cos\theta_2;\ v + Z_2\tau = \lambda_2 \sin\theta_2 \end{cases} \tag{1}$$

Eliminating λ_1 and λ_2, we obtain two linear constraints in $[u, v, \sigma, \tau]$ as

$$\begin{cases} \sigma &= (Au + Bv)/E \\ \tau &= (Cu + Dv)/E \end{cases} \tag{2}$$

where

$$A = Z_2 \cos\theta_2 \sin\theta_1 - Z_1 \cos\theta_1 \sin\theta_2, \quad B = (Z_1 - Z_2) \cos\theta_1 \cos\theta_2,$$
$$C = (Z_1 - Z_2) \sin\theta_1 \sin\theta_2, \quad D = Z_1 \cos\theta_2 \sin\theta_1 - Z_2 \cos\theta_1 \sin\theta_2,$$
$$E = Z_1 Z_2 \sin(\theta_2 - \theta_1)$$

We call Eqn (2) the XSlit ray constraints that maps a pixel (u, v) to a ray with direction $(\sigma, \tau, 1)$

Appendix B: XSlit Point Projection

Consider a 3D point $\mathbf{X} = (x, y, z)$. For each ray $[u, v, \sigma, \tau]$ passing through $\mathbf{X}$, there exist some λ that satisfies

$$[u, v, 0] + \lambda[\sigma, \tau, 1] = [x, y, z] \tag{3}$$

It's easy to see that $\lambda = z$. By eliminating λ, we have

$$\begin{cases} u + z\sigma = x \\ v + z\tau = y \end{cases} \tag{4}$$

Combining Eqn. (4) with the XSlit ray constraints (Eqn. (2)), we can solve (u, v) w.r.t.$\mathbf{X}$ as

$$\begin{cases} u = \dfrac{(Dx - By)Ez + E^2x}{(AD - BC)z^2 + (A + D)Ez + E^2} \\ v = \dfrac{(Ay - Cx)Ez + E^2y}{(AD - BC)z^2 + (A + D)Ez + E^2} \end{cases} \tag{5}$$

We call Eqn (5) the *point projection equation* in XSlit cameras.

Appendix C: XSlit Line Projection

For a line l in the scene not parallel to the image plane, we can parameterize it under 2PP as $[u_l, v_l, \sigma_l, \tau_l]$. For each ray $[u, v, \sigma, \tau]$ passing through l, there must exist some λ and λ_l so that

$$[u, v, 0] + \lambda[\sigma, \tau, 1] = [u_l, v_l, 0] + \lambda_l[\sigma_l, \tau_l, 1] \tag{6}$$

We have $\lambda = \lambda_l$ and by eliminating λ andλ_l, we obtain a bilinear constraint

$$\frac{u - u_l}{v - v_l} = \frac{\sigma - \sigma_l}{\tau - \tau_l} \tag{7}$$

By substituting σ and τ with u and v using Eqn. (2), we obtain a conic curve in u and v, *i.e.*, the image of l as

$$\begin{aligned} Cu^2 + (D - A)uv - Bv^2 + (Av_l - Cu_l - E\tau_l)u \\ + (Bv_l - Du_l + E\sigma_l)v + E(u_l\tau_l - v_l\sigma_l) = 0 \end{aligned} \tag{8}$$

We call Eqn (8) the *line projection equation* in XSlit cameras. To determine the type of the conic, we compute its determinant

$$J = (D-A)^2 - 4BC = \sin^2(\theta_2 - \theta_1)(Z_1 - Z_2)^2 > 0 \tag{9}$$

Therefore, the conic can only be hyperbolas in an XSlit camera.