

# The Light-Path Less Traveled

Srikumar Ramalingam<sup>1</sup> Sofien Bouaziz<sup>2</sup> Peter Sturm<sup>3</sup> Philip H. S. Torr<sup>4</sup>

<sup>1</sup>Mitsubishi Electric Research Lab (MERL), Cambridge, USA

<sup>2</sup>Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland

<sup>3</sup>INRIA Grenoble – Rhône-Alpes and Laboratoire Jean Kuntzmann, Grenoble, France

<sup>4</sup>Oxford Brookes University, Oxford, UK

## Abstract

*This paper extends classical object pose and relative camera motion estimation algorithms for imaging sensors sampling the scene through light-paths. Many algorithms in multi-view geometry assume that every pixel observes light traveling in a single line in space. We wish to relax this assumption and address various theoretical and practical issues in modeling camera rays as piece-wise linear-paths. Such paths consisting of finitely many linear segments are typical of any simple camera configuration with reflective and refractive elements. Our main contribution is to propose efficient algorithms that can work with the complete light-path without knowing the correspondence between their individual segments and the scene points. Second, we investigate light-paths containing infinitely many and small piece-wise linear segments that can be modeled using simple parametric curves such as conics. We show compelling simulations and real experiments, involving catadioptric configurations and mirages, to validate our study.*

## 1. Introduction and motivation

The bending of light rays is a very common natural phenomenon producing very many optical effects; reflection on water, refraction in a dew drop on a leaf, distortion of underwater objects, shimmering on a road's surface, the blue oasis in the desert, rainbows, lingering sunset, halo surrounding the sun and twinkling stars are just a few examples. Despite the significant progress made by the vision and graphics communities toward realistic models, we are still far from modeling the extreme complexity of light.

Many algorithms in multi-view geometry use either the pinhole model, where light rays pass through a single optical center, or a non-central model where every pixel is mapped to an arbitrary projection ray. Non-central camera models have been studied in the context of catadioptric configurations [9, 17, 21, 28] and most multi-view geometry algorithms have been extended to such models. Most of these algorithms assume that a pixel is mapped to a sin-

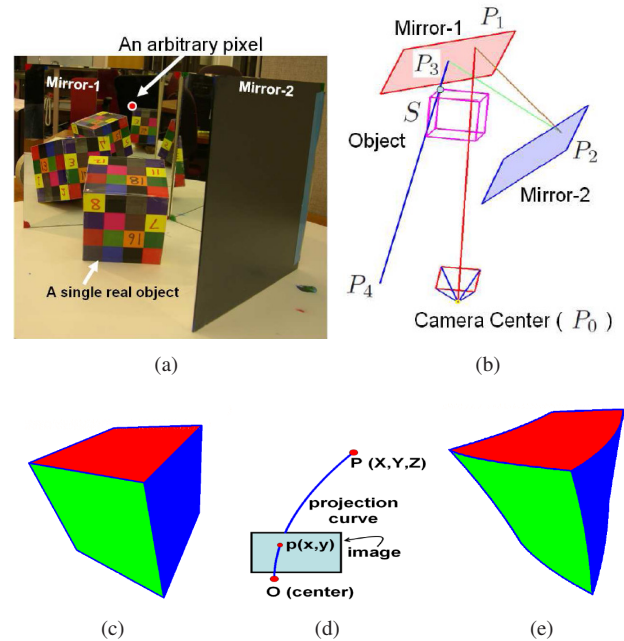


Figure 1. A setup with three planar mirrors and a camera facing two of them is shown in (a). The light-path for a chosen pixel is shown in (b). The main highlight of this paper is to extend the classical pose and motion estimation algorithms for such paths without any prior knowledge about the correspondence between the scene point and the individual segment in the piece-wise linear paths. We show a cube imaged using a pinhole camera in (c). By sampling light along a parametric curve shown in (d), we synthesize the same cube in (e).

gle straight line in space. We wish to relax this assumption and associate several piece-wise linear segments to a single pixel and propose pose and motion estimation algorithms. In particular we wish to study this problem without using any prior knowledge about the segment of the light-path interacting with the scene. One may wonder: Why is this generality necessary? Consider Figure 1(a) where we show a simple configuration with three mirrors and a camera. After two or three bounces with two mirrors, it becomes extremely difficult to know the correspondence between the

scene point and its associated segment in the light-path.

There is a wide body of literature on reconstruction algorithms involving specular objects [4, 30] (reflective and refractive)<sup>1</sup>. In specular stereo works [3, 20], the path of the light ray, before and after reflection, is studied to recover the shape of mirror-like objects from two views. Ben-Ezra and Nayar detect and reconstruct transparent objects from a sequence of images taken under known motion [1]. Here, a physics based modeling approach is taken to handle refraction and to reconstruct the shape of transparent objects in the form of super-ellipsoids. Kutulakos and Steger have shown some inspiring results in reconstructing specular objects by recovering the path of a light ray after it undergoes refraction [16]. In order to recover the light-path they use reference 3D points whose coordinates are known with respect to the camera. Although we also recover the light-paths accurately using non-trivial techniques for investigation purposes, our main contribution is not the light-path computation. Rather, we search for scene points on the different segments of the known light-path using pose estimation and motion estimation algorithms. Chari and Sturm developed geometric entities like fundamental matrix for underwater scenarios [5]. Seitz et al. [25] have investigated the multi-bounce nature of light-paths for decomposing images and removing inter reflections. Recently Kirmani et al. [14] used multi-path analysis of light transport to reconstruct the geometry of hidden regions, which are not in the line of sight of the camera.

In this paper, we also investigate algorithms for light-paths containing infinitely many and small piece-wise linear segments that can be modeled using conics. Although we are not familiar with any prior work for optical images, Hartley and Saxena have used curved projection rays for modeling SAR imagery [13].

We summarize our main contributions below:

- We develop pose and motion estimation algorithms for cameras where each pixel samples light traveling in a piece-wise linear path or a parametric curve. We refer to these cameras as piece-wise linear model (PLM) and parametric curve model (PCM).
- We show that the correspondence problem between a scene point and the individual segments of the piece-wise linear path can be mapped to the enumeration of all the maximum cliques in an associated graph.
- The main contribution of this paper is an efficient algorithm for PLMs that can work with a large number of piece-wise segments. We propose an extremely useful pairwise cheirality constraint that allows one to search in the large solution space to solve light-paths with finitely many segments. In particular, we show that it

is possible to extend the pose estimation for light-paths having more than 100 segments.

- We show compelling simulations and real experiments to validate our theory for PLMs and PCMs. To work with real images of mirages, we include a practical method to compute the refraction parameters from an image of a mirage.

**Overview of the paper:** In section 2 we introduce and develop pose and motion estimation algorithms for PLMs. In section 3 we propose an efficient search algorithm for finding the correspondence between the individual segment of a piece-wise linear light-path and a scene point. In section 4 we introduce and develop multi-view geometry algorithms for PCMs. In section 5 we show simulations and real experiments to validate our theory. We use simple camera configurations with planar mirrors to show the results for PLMs and use real images of mirages to demonstrate the results for PCMs.

## 2. The piece-wise linear model

Every pixel samples light in a piece-wise linear path denoted by a sequence of 3D points  $P_0, P_1, \dots, P_n$  where  $n$  is the number of segments. We refer to this path as PLP. In Figure 1(a), we show a configuration consisting of three planar mirrors and a camera facing two of them. The camera is facing the back-side of one of the mirrors. For a chosen pixel shown on the image, we trace the corresponding PLP. It consists of the segments  $P_4P_3$ ,  $P_3P_2$ ,  $P_2P_1$  and  $P_1P_0$ . In a PLM all pixels are associated to such paths and we assume that they are pre-calibrated. We refer to an object or the scene as not being part of the PLM. As an object enters the field of view of a PLM, every point on the object will lie on multiple segments in various PLPs. This is the reason for observing the same 3D point at multiple places in the image. However, for a given pixel, the corresponding scene point lies in general on only one segment of the pixel's PLP. For example the scene point  $S$  corresponding to the chosen pixel resides on the segment  $P_4P_3$ . In general it is not easy to identify this association between the segment and 3D point even manually when there are multiple segments in a light-path. We explore the feasibility of finding this correspondence automatically while we solve the pose and motion estimation problems.

### 2.1. Pose Estimation

Given three correspondences between points in the world and their projections on the images, the goal is to compute the pose of the camera in the world coordinate system. For the pinhole model, many solutions have been proposed in the literature - Grunert [10], Fischler and Bolles [8], Church's method [6], Haralick et al. [11], to name but a few references. One can also compute pose using both

<sup>1</sup>We do not address the problem of diffusion where a single light ray may get split into infinitely many rays.

points and lines [22], but in this paper we focus on only points. Recently, there has been algorithms for developing pose estimation using three points for non-central or generalized cameras [19, 23]. In a generalized camera model, every pixel is mapped to a projection ray in space along which it samples light [9, 27]. Mathematically, the minimal pose estimation problem is described as follows. Given 3 points and 3 rays in different coordinate frames, find a rigid transformation such that the points are incident with their corresponding rays. In this problem, the number of rays and points is minimal for computing the transformation. This algorithm gives 8 solutions in general and additional correspondences are used to prune the ones inconsistent with the other matches. This algorithm is generally employed in a hypothesize-and-test framework such as RANSAC [8]. We use this algorithm as the basic block for developing ours.

We briefly describe the pose estimation problem for PLMs. Given three scene points and their corresponding pixels, thus their corresponding PLPs, our goal is to compute a transformation such that each point lies on one of the segments in its corresponding PLP. Once the correspondence between points and line segments is established, we may compute the pose using the above generalized pose estimation algorithm. A correct transformation could be easily verified by checking if at least one segment of every PLP contains the corresponding scene point. Thus the remaining missing block in developing a pose estimation algorithm is to compute the correspondence between the segments in a PLP and its corresponding point. One can use a brute force search strategy to generate a lot of poses and identify the correct pose from them. However, this is infeasible when there are many segments in each PLP. In section 3 we propose an efficient search strategy to solve the pose estimation problem using a pairwise cheirality constraint. Without this constraint, the exhaustive search is highly infeasible.

## 2.2. Motion Estimation

The underlying mathematical problem for generalized motion estimation is briefly described here: Given two sets of 6 rays each, the goal is to rotate and translate one set such that every ray in one set intersects its corresponding ray in the other. Stewénius et al. gave the solution for generalized cameras that leads to 64 solutions [26]. We can use additional correspondences to prune the ones that are inconsistent with other matches. This algorithm will be used as the basic block for developing the motion estimation algorithm for PLMs.

We briefly describe the motion estimation problem for PLMs. Given correspondences between two sets of 6 PLPs in two cameras, the goal is to compute a transformation such that one segment in a PLP from the first set intersects with at least one segment in its corresponding PLP in the second set. Once the correspondence between the segments in every pair of PLP is established, we may compute the

motion using the above generalized motion estimation algorithm. A correct transformation could be easily verified by checking if at least one segment of every PLP intersect at least one segment of its corresponding PLP. Similar to the pose estimation problem, we could employ a brute force search to generate all possible correspondences. However, this is even harder than the pose problem. Using a pairwise cheirality constraint we reduce the search space and solve the motion estimation.

## 3. The correspondence problem

We describe the correspondence problem for the pose estimation problem in detail. The algorithm for motion estimation can be analogously developed. In order to efficiently solve the correspondence problem we primarily use one geometric constraint. This constraint is a pairwise one where we can jointly check whether two point-PLP correspondences can jointly hold true. In Figure 2(a),  $\mathcal{L}_P$  and  $\mathcal{M}_Q$  are two segments from two different PLPs. In order for the 3D points  $P$  and  $Q$  to correspond to segments  $\mathcal{L}_P$  and  $\mathcal{M}_Q$ , the following condition must be satisfied:

$$d_{\mathcal{L}_P, \mathcal{M}_Q}^{\min} \leq d_{P, Q} \leq d_{\mathcal{L}_P, \mathcal{M}_Q}^{\max} \quad (1)$$

where  $d_{\mathcal{L}_P, \mathcal{M}_Q}^{\min}$  and  $d_{\mathcal{L}_P, \mathcal{M}_Q}^{\max}$  are the minimum and maximum Euclidean distances between the line segments  $\mathcal{L}_P$  and  $\mathcal{M}_Q$ , and  $d_{P, Q}$  is the distance between the 3D points  $P$  and  $Q$ . Later, we will observe that the above simple geometric constraint reduces the search space for the pose problem significantly. We refer to this as pairwise cheirality due to its resemblance to the classical cheirality constraint [12]. The classical one says that the scene points must lie in front of the camera that view them.

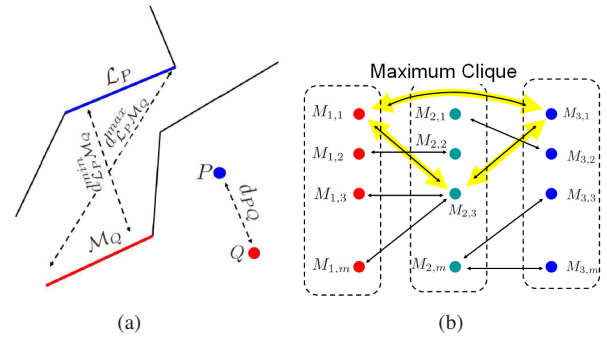


Figure 2. (a) We show the pairwise cheirality constraint for the problem of pose estimation for PLMs. (b) The solution space for correspondences between the individual line segments of a PLM and the corresponding 3D points could be mapped to the enumeration of all possible maximum cliques of size 3 in a tri-partite graph.

We will now describe a method to compute the possible correspondences satisfying the pairwise cheirality constraint. Consider the graph shown in Figure 2(b). Every

node  $M_{ij}$  represents the correspondence between the  $i$ th point and the  $j$ th segment in the corresponding PLP. An edge between  $M_{ij}$  and  $M_{kl}$  exists if two pairwise assignments can happen simultaneously without conflicting with each other. This implies that the pairwise assignment satisfies the pairwise cheirality constraint given in equation 1 and the uniqueness constraint. The uniqueness constraint refers to the rule that the same 3D point cannot lie on two different segments of the same PLP. Thus there is no edge between any  $M_{ij}$  and  $M_{ik}$ . All the candidate solutions are given by the maximum cliques of the graph. The maximum clique of a graph refers to the largest complete subgraph, i.e. cliques, where every pair of nodes are connected by an edge. Note that all the maximum cliques have size three because the graph is tri-partite. Thus we identify triplet of nodes or correspondences, where every pair of nodes is consistent. If each PLP has  $n$  segments, the brute force approach leads to  $n^3$  candidates whereas our approach leads to a much lower number of candidates as shown in section 5.

In the case of motion estimation we can have a similar pairwise cheirality constraint for pairwise assignments. Let us assume that for the correct transformation the pairs of segments  $(\mathcal{L}, \mathcal{M})$  and  $(\mathcal{R}, \mathcal{S})$  intersect each other. The intersection is only possible if the following two conditions hold true:

$$d_{\mathcal{L}, \mathcal{M}}^{\min} < d_{\mathcal{R}, \mathcal{S}}^{\max} \quad \text{and} \quad d_{\mathcal{R}, \mathcal{S}}^{\min} < d_{\mathcal{L}, \mathcal{M}}^{\max} \quad (2)$$

Similar to the pose estimation problem, the correspondence problem for motion estimation can be mapped to the enumeration of maximum cliques of size 6 in a 6-partite graph. These results are not entirely surprising because other correspondence problems in computer vision have been mapped to similar NP-hard problems before [7, 29]. For each candidate match, we compute the motion using generalized motion estimation algorithm and the correct solution is identified using additional PLPs.

## 4. The parametric curve model

In the previous section we observed that the solution space increases exponentially with the number of segments in each PLP. In several natural phenomena, the light-path contains infinitely many small piece-wise segments as in the case of mirages. For such light-paths, the algorithm for PLM is infeasible. Here we show that despite the infinitely many segments, pose and motion estimation algorithms are feasible if the segments fit a simple parametric curve like a conic. In order to do this we will represent the light-path using a parametric curve, generally represented as  $(x = x(t), y = y(t), z = z(t))$ , where  $t$  is an independent parameter which helps us to navigate along the path of the curve. A general polynomial parametric curve can be

given by:

$$x(t) = \sum_{i=0}^n a_i t^i, \quad y(t) = \sum_{i=0}^n b_i t^i, \quad z(t) = \sum_{i=0}^n c_i t^i, \quad (3)$$

where  $a_i$ ,  $b_i$  and  $c_i$  are coefficients in the curve. In what follows, we consider a simple parametric representation to illustrate the basic ideas. We assume that the curves pass through the optical center and the nonlinearity is only along the  $x$  dimension. In the experiments, we use a similar model for mirages. A parametric curve path (PCP) can be represented using the following form:

$$\begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix} = \begin{pmatrix} at^2 + bt \\ ct \\ t \end{pmatrix} \quad (4)$$

where  $a, b$  and  $c$  are the parameters of the curve. By varying the parameter  $t$  from 0 to  $\infty$  we can navigate along the path of the curve. Such a curve is conic-shaped and an example is shown in figure 1(d).

### 4.1. Motion Estimation

Given two sets of corresponding projection curves from two cameras, the goal is to compute a transformation such that every projection curve intersects its counterpart. This also means that there exists a common point on both the projection curves if they are expressed in a common reference frame. For simplicity, we assume that one of the camera is a PCM and the other one is a classical pinhole camera. A parametric representation of the classical pinhole ray is given below:

$$\begin{pmatrix} x(t_2) \\ y(t_2) \\ z(t_2) \end{pmatrix} = \begin{pmatrix} a_2 t_2 \\ b_2 t_2 \\ t_2 \end{pmatrix} \quad (5)$$

Under a general motion of  $(\mathbf{R}, \mathbf{T})$  we have the following equation:

$$\begin{pmatrix} a_1 t_1^2 + b_1 t_1 \\ c_1 t_1 \\ t_1 \end{pmatrix} = \mathbf{R} \begin{pmatrix} a_2 t_2 \\ b_2 t_2 \\ t_2 \end{pmatrix} + \begin{pmatrix} T_1 \\ T_2 \\ T_3 \end{pmatrix} \quad (6)$$

Applying algebraic transformations we eliminate  $t_1$  and  $t_2$  and we get the equation  $\sum_{i=1}^{36} C_i V_i = 0$  where  $C_i$  are functions that depend on the known parameters  $a_1, b_1, c_1, a_2$  and  $b_2$ . The coupled variables  $V_i$  are functions of the unknown motion parameters.

The coupled variables can be estimated using singular value decomposition. Using orthogonality constraints on the rotation matrix the individual motion parameters can be extracted from the coupled variables. The above method needs at least 35 point correspondences, although the number of independent degrees of freedom is only 6. One can also solve the pose estimation problem in a similar manner and it is much simpler.



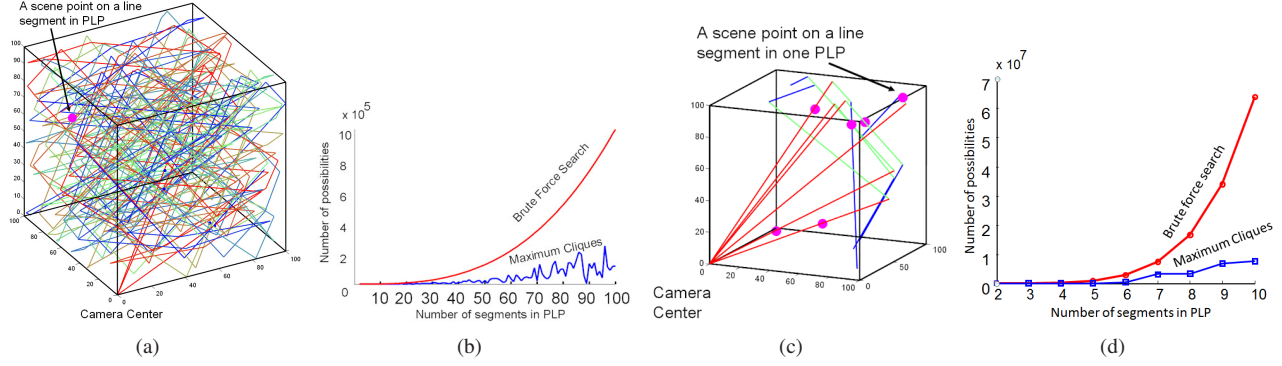


Figure 3. We show the simulation platform in (a) and (c). A single PLP is traced to a path containing 100 segments in (a). The PLP is color-coded from red to blue, i.e., the initial segments are red and the final ones are blue. In (b) we show the comparison between brute force search and our maximum clique algorithm for pose estimation. In (c) we show 6 PLPs with three segments each. In (d) we show the comparison between brute force search and our algorithm for generating the candidate solutions for motion estimation. [best viewed in color]

## 5. Experiments

**Simulations for PLM:** We show the simulation platform in Figure 3. The camera is at one of the corners of a cube of dimension 100. The camera is facing the inside of the cube whose six walls are reflective. Using a simple ray-tracing technique we obtain light-paths with as many segments as possible starting with random directions from the origin. This allows us to generate PLPs of any length for a given pixel in the image with known calibration parameters. In order to simulate the pose estimation experiment, we generated random PLPs each of length  $n$ . The PLP for a single pixel having 100 segments<sup>2</sup> is shown in Figure 3(a). We select a random segment in each PLP and select a random point on it. We generate one 3D point for every PLP. These points were rotated and translated by a random transformation matrix. We used our search strategy to generate all the candidate solutions. In Figure 3(b) we show the reduction in solution space using our maximum clique formulation. Using 4 or more points, we were able to compute the exact pose from PLPs having as many as 100 segments. The algorithm identified the correct segments even in the presence of noise. Note that only the candidate matches are generated using the minimal set of 3 correspondences. For each candidate match, we compute the pose using generalized pose estimation algorithm and check its validity using additional points. The computed pose is checked using a few other points. In general 4 points were sufficient to identify the correct pose. The solution space for pose estimation is  $n^3$  using a brute force search. The number of candidates obtained by the enumeration of the maximum cliques is much lower, as shown in Figure 3(b).

We used a similar simulation platform for testing the motion estimation algorithm. It is difficult to generate matches,

since this involves finding PLPs in two cameras that intersect each other, which is a hard problem. In the first camera, we generate PLPs of arbitrary length. Seven or more random points were chosen on random segments in each PLP. These points were rotated and translated by a random transformation matrix. For the second camera we used a single bounce to reach the camera center. In other words, we were able to compute PLPs for the second camera of length 2. Note that the solution space of motion estimation is extremely large compared to the pose estimation problem. For two sets of PLPs, each of length  $n$ , the overall solution space using brute force search is  $n^{12}$ . Using our maximum clique formulation we reduced the search space significantly.

**Real Experiments for PLM:** We show results for real experiments with the setup shown in Figure 4. Testing was done using two mirrors and considering light-paths up to length 3. The planar mirrors are squares of size 304 mm each. The cube is of size 127 mm. The correspondences are given manually in the images. Note that automatic matching algorithms that are invariant to mirror flips could also be used to match point features. We used images of size  $2272 \times 1704$ . The main challenge was the precise calibration of the two mirrors. Using the initial calibration of the mirrors we did a bundle adjustment from known calibration grid points after multiple reflections. This improved the orientation of the planes significantly and the setup was precise enough for our experiments. We believe that this idea of optimizing on the light-paths after multiple reflections would prove useful for several other catadioptric configurations. After computing the motion, we reconstructed the cube by intersecting the correct segments of the matching PLPs. We measured the error using the difference between the ground-truth data and the reconstructed model. The overall RMS errors in pose and 3D reconstruction results are 8 mm and

<sup>2</sup>The experimental setup is inspired by the illumination problem, where light-paths are indefinitely traced inside a mirror-walled polygon.

15 mm respectively. Note that this error is without any further refinement using bundle adjustment.

**Degeneracy:** Axial cameras refer to a class of cameras where all the projection rays intersect in a single line in space [24]. During the experiments, we observed that the axial configuration might be a degenerate case for the 6-point motion estimation algorithm [26]. In our real and synthetic experiments, we did not use an axial configuration.

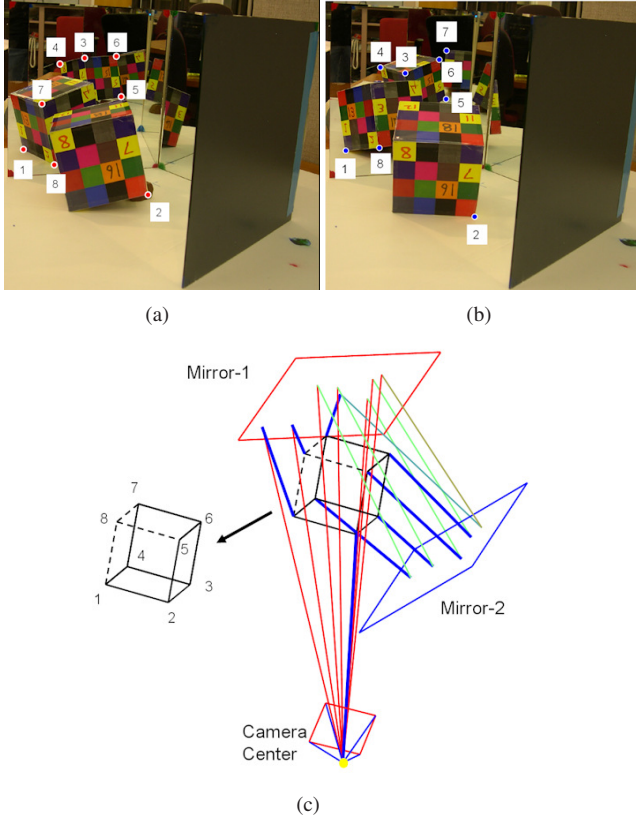


Figure 4. (a) and (b) are two images captured from the same pre-calibrated setup using a single camera and two mirrors. The cube is rotated and translated between the two views. The correspondences are given manually. Every 3D point occurs multiple times in the images. Our algorithm can work with any correct match. The reconstructed cube and the PLPs are shown in (c). The correct segment is automatically identified and it is shown in blue. [best viewed in color]

**Modeling PCMs using conics:** We simulated a PCM whose projection curves are conic-shaped. Images of 3D synthetic models of a unit cube are generated using non-linear ray tracing. Algorithms such as motion estimation and 3D reconstruction were tested. The concept of epipolar lines generalizes to epipolar curves. Analogously epipolar planes manifest themselves as epipolar surfaces. The basic idea is very simple; for every pixel  $x$  in the left image we look at all the pixels in the right image whose rays intersect the ray associated with pixel  $x$ . In figure 5, the synthesized

images and the epipolar curves on both the PCM and the pinhole images are shown.

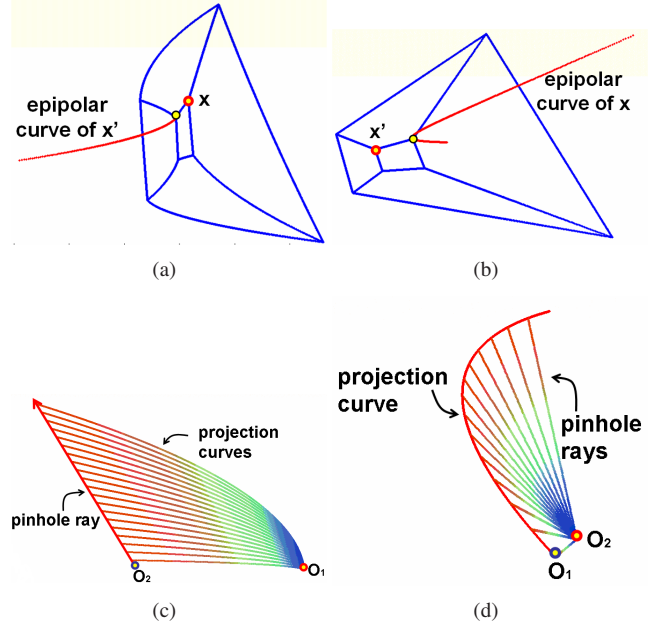


Figure 5. (a) and (b) show synthesized images of the inside of a cube using a PCM and a pinhole respectively. (c) Epipolar curve on the PCM image is shown. This curve is obtained by collecting the pixels whose projection rays (conic-shaped rays) intersect a projection ray (a straight line) of pixel  $x'$  in the pinhole image. (d) The epipolar curve on the pinhole image is the set of pixels whose projection rays intersect the conic-shaped ray of pixel  $x$  in the PCM image.

**Simulation of Mirages:** We use a physics based modeling for mirages [2, 18]. Mirages occur when the refractive index of air changes from one region to another. When a light ray passes from one medium to another having different refractive indices, its path is determined by Snell's law. In our modeling the atmosphere is sliced into very small layers of constant refractive index each and we use Snell's law at each boundary. We show the path of a mirage ray in figure 6. The light ray enters at an angle of  $\theta$  with respect to the vertical. Let  $n$  be the refractive index at the starting point ( $y = 0$ ). Let  $m$  refer to the parameter that describes the variation in refractive index relative to altitude. Thus the refractive index for a specific altitude  $y$  is  $n + my$ . The parametric model for the light rays during the formation of mirages is given below:

$$\begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix} = \begin{pmatrix} c \\ t \\ f(t, k, m, n) \end{pmatrix} \quad (7)$$

where

$$f(t, k, m, n) = \frac{k}{m} \log(mt + n + \sqrt{(mt + n)^2 - k^2}) - \frac{k}{m} \log(n + \sqrt{n^2 - k^2}) \quad (8)$$

where  $c$  refers to a constant and  $k$  is given by  $n \sin(\theta)$ . The above equation refers to the mirage ray corresponding to one pixel in the image. We assume that all mirage rays corresponding to pixels in one vertical scanline in the image lie on a single plane.

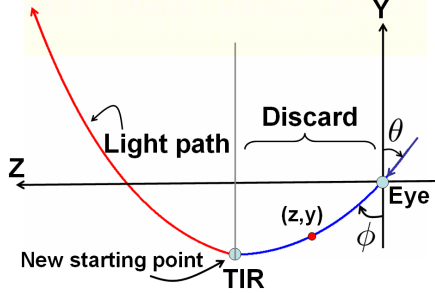


Figure 6. A single light curve during the formation of a mirage. Note that the curve undergoes total internal reflection (TIR) and changes its path.

During the formation of a mirage, total internal reflection (TIR) also takes place as shown in figure 6. We refer to the point at which TIR happens as the *turning point*. When a light ray passes into a less dense medium and the angle of incident light is greater than the critical angle, TIR takes place. In our work, we compute the turning point, shift the origin to this point, and finally update the incident angle. In other words, the initial part of the ray is not useful to model the mirages. Thus we remove this part and model the remaining projection curve. Using Taylor expansion, one can show that the above model for mirages shown in Equation 8 can be approximated by the following parametric curve:

$$\begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix} = \begin{pmatrix} f \\ dt + e \\ at^2 + bt + c \end{pmatrix} \quad (9)$$

where  $a, b, c, d, e$  and  $f$  are known constants that describe the path of the light ray. Note that the above approximation is valid for typical values of  $m$ , which is very small for normal temperature variations.

**Mirage simulation and testing:** We test the accuracy of our modeling using a synthesize-and-compare algorithm. For a given image of the mirage we approximate the different objects in the image as planar segments. For example in Figure 8, the desert image has two types of planes: vertical planes corresponding to the trees and a horizontal plane corresponding to the ground. We use different depth values for the trees and synthesize mirages by non-linear ray-tracing [2, 18]. We synthesize mirages for different values of the calibration parameters: change in refraction index ( $m$ ), field of view and scaling along  $z$  and  $y$  axis. For example in Figure 7, we show the synthesis of mirages for different values of  $m$ . In order to do the synthesis we used a pixel shader program in GPU to generate several mirages per second and

do the matching. By doing chamfer matching between the synthesized and real images of the mirages we were able to optimize on the chosen parameters. Some of the results based on this matching are shown in figure 8. Testing the motion estimation and pose estimation algorithms for real images of mirages are very challenging and they are still unresolved problems. The main bottleneck is the lack of real data. In order to test motion estimation, we need two images of mirages of the same scene from different view-points along with the calibration parameters. The pose estimation requires a known object in the scene along with the mirage parameters.

## 6. Discussion

We made a few non-trivial observations of light-paths using theoretical analysis, simulations and real experiments. We still believe that the experiments do not convey the generality of the proposed technique. Note that once the light-paths are known, our techniques can apply to several scenarios like refraction through water or glass and reflections on arbitrary shaped mirrors. Computation of long light-paths (3 or more segments) in these scenarios is an active area of research and involve specialized calibration techniques, which is not the main focus of this paper. We believe that the ability to compute pose and motion in such challenging scenarios will spur further research in exact light-path computation techniques and their utilization in applications.

We used simple optimization techniques to study PLMs and PCMs. For example, we enumerated all the maximum cliques in a graph using a simple tree-based search. We could use efficient branch and bound techniques to improve the performance. The use of Groebner basis solvers [15] might also prove beneficial for modeling higher degree curves in PCMs.

**Acknowledgments:** Srikumar Ramalingam would like to thank Jay Thornton for the support. We would like to thank Matthew Brand, Yuichi Taguchi, Amit Agrawal and Visesh Chari for useful discussions on light-paths.

## References

- [1] M. Ben-Ezra and S. Nayar. What does motion reveal about transparency? In *ICCV*, 2003.
- [2] M. Berger, T. Trout, and N. Levit. Raytracing mirages. *IEEE Computer Graphics and Applications*, 1990.
- [3] T. Bonfort and P. Sturm. Voxel carving for specular surfaces. In *ICCV*, 2003.
- [4] G. D. Canas, Y. Vasilyev, Y. Adato, T. Zickler, S. Gortler, and O. Ben-Shahar. A linear formulation of shape from specular flow. In *ICCV*, 2009.
- [5] V. Chari and P. Sturm. Multi-view geometry of the refractive plane. In *BMVC*, 2009.
- [6] C. S. (editor). *Manual of Photogrammetry*. Fourth Edition, ASPRS, 1980.



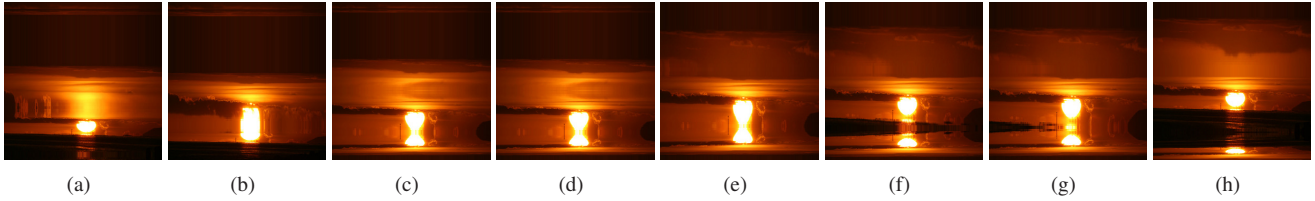


Figure 7. A realistic mirage video generated with our program by varying the parameter  $m$  from 0.0044 to 0.0067.

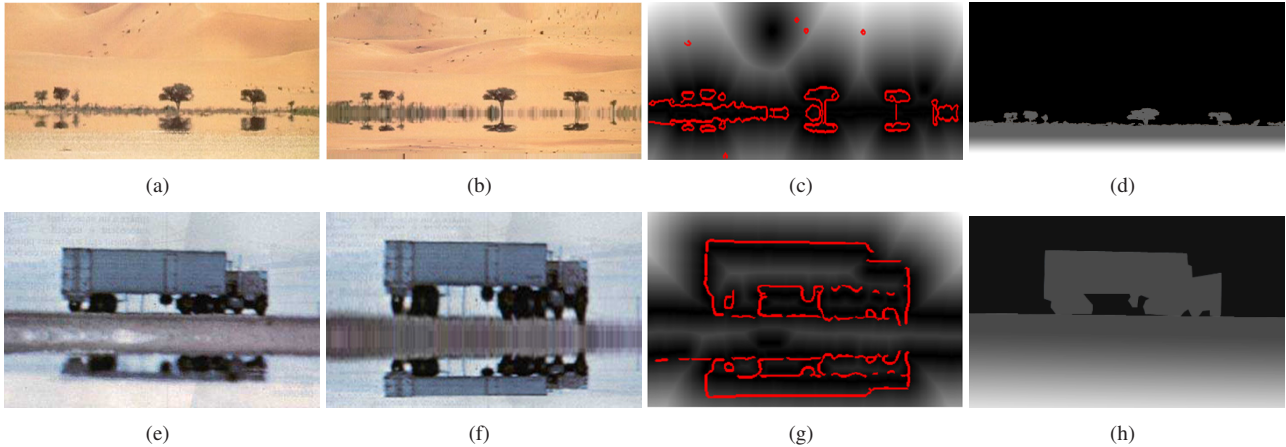


Figure 8. (a, e) Real images of mirages. (b, f) Synthesis of mirages using non-linear ray tracing. (c, g) The chamfer matching between the real and the synthesized images to compute the mirage parameters. (d, h) Estimated 3D depth under planar assumptions of the different layers in the scene.

- [7] O. Enqvist, K. Josephson, and F. Kahl. Optimal correspondences from pairwise constraints. In *ICCV*, Sept. 2009.
- [8] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981.
- [9] M. Grossberg and S. Nayar. A general imaging model and a method for finding its parameters. In *ICCV*, 2001.
- [10] J. Grunert. Das pothenotische Problem in erweiterter Gestalt nebst über seine Anwendungen in der Geodäsie. *Grunerts Archiv für Mathematik und Physik*, 1:238248, 1841.
- [11] R. Haralick, C. Lee, K. Ottenberg, and M. Nolle. Review and analysis of solutions of the three point perspective pose estimation problem. *IJCV*, 1994.
- [12] R. Hartley. Cheirality invariants. In *DARPA Image Understanding Workshop*, 1993.
- [13] R. Hartley and T. Saxena. The cubic rational polynomial camera model. In *DARPA Image Understanding Workshop*, pages 649–653, 1997.
- [14] A. Kirmani, T. Hutchison, J. Davis, and R. Raskar. Looking around the corner using transient imaging. In *ICCV*, 2009.
- [15] Z. Kukelova, M. Bujnak, and T. Pajdla. Automatic generator of minimal problem solvers. In *ECCV*, 2008.
- [16] K. Kutulakos and E. Steger. A theory of refractive and specular 3d shape by light-path triangulation. In *ICCV*, 2005.
- [17] B. Micusik and T. Pajdla. Autocalibration and 3d reconstruction with non-central catadioptric cameras. In *CVPR*, 2004.
- [18] F. K. Musgrave and M. Berger. A note on ray tracing mirages. *IEEE Computer Graphics and Applications*, 1990.
- [19] D. Nistér. A minimal solution to the generalized 3-point pose problem. In *CVPR*, 2004.
- [20] M. Oren and S. Nayar. A theory of specular surface geometry. In *ICCV*, 1995.
- [21] R. Pless. Using many cameras as one. In *CVPR*, 2003.
- [22] S. Ramalingam, S. Bouaziz, and P. Sturm. Pose estimation using points and lines for geo-localization. In *ICRA*, 2011.
- [23] S. Ramalingam, S. Lodha, and P. Sturm. A generic structure-from-motion framework. In *CVIU*, 2006.
- [24] S. Ramalingam, P. Sturm, and S. Lodha. Theory and calibration algorithms for axial cameras. In *ACCV*, 2006.
- [25] S. Seitz, Y. Matsushita, and K. Kutulakos. A theory of inverse light transport. In *ICCV*, 2005.
- [26] H. Stewenius, D. Nister, M. Oskarsson, and K. Astrom. Solutions to minimal generalized relative pose problems. In *OMNIVIS*, 2005.
- [27] P. Sturm and S. Ramalingam. A generic concept for camera calibration. In *ECCV*, volume 2, pages 1–13, 2004.
- [28] P. Sturm, S. Ramalingam, J.-P. Tardif, S. Gasparini, and J. Barreto. Camera models and fundamental concepts used in geometric computer vision. *Foundations and Trends in Computer Graphics and Vision*, 2011.
- [29] P. Tu, T. Saxena, and R. Hartley. Recognizing objects using color-annotated adjacency graphs. In *Lecture Notes in Computer Science: Shape, Contour and Grouping in Computer Vision*, 1999.
- [30] D. E. Zongker, D. M. Werner, B. Curless, and D. H. Salesin. Environment matting and compositing. In *SIGGRAPH*, 1999.