Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computer Vision and Image Understanding 97 (2005) 51–85
Computer Vision and Image Understanding
www.elsevier.com/locate/cviu

Extracting layers and analyzing their specular properties using epipolar-plane-image analysis

Antonio Criminisi[a,*], Sing Bing Kang[a], Rahul Swaminathan[b], Richard Szeliski[a], P. Anandan[a]

[a] *Microsoft Corporation, Cambridge, UK*
[b] *Columbia University, USA*

Received 11 June 2003; accepted 4 June 2004
Available online 7 August 2004

Abstract

Despite progress in stereo reconstruction and structure from motion, 3D scene reconstruction from multiple images still faces many difficulties, especially in dealing with occlusions, partial visibility, textureless regions, and specular reflections. Moreover, the problem of recovering a *spatially dense* 3D representation from many views has not been adequately treated. This document addresses the problems of achieving a dense reconstruction from a sequence of images and analyzing and removing specular highlights. The first part describes an approach for automatically decomposing the scene into a set of spatio-temporal layers (namely *EPI-tubes*) by analyzing the epipolar plane image (EPI) volume. The key to our approach is to directly exploit the high degree of regularity found in the EPI volume. In contrast to past work on EPI volumes that focused on a sparse set of feature tracks, we develop a complete and dense segmentation of the EPI volume. Two different algorithms are presented to segment the input EPI volume into its component EPI tubes. The second part describes a mathematical characterization of specular reflections within the EPI framework and proposes a novel technique for decomposing a static scene into its diffuse (Lambertian) and specular components. Furthermore, a taxonomy of specularities based on their photometric properties is presented as a guide for designing further separation techniques. The validity of our approach is demonstrated on a number of sequences of complex scenes with large amounts of occlusions

* Corresponding author. Fax: +44-(0)1223-479999.
E-mail address: antcrim@microsoft.com (A. Criminisi).

1077-3142/$ - see front matter
doi:10.1016/j.cviu.2004.06.001

and specularity. In particular, we demonstrate object removal and insertion, depth map estimation, and detection and removal of specular highlights.

1. Introduction

Despite progress in stereo reconstruction and structure from motion, 3D scene reconstruction from multiple images still faces many difficulties, especially in dealing with occlusions, partial visibility, and textureless regions. While the problem of multi-view scene reconstruction from feature correspondences has been extensively studied [11], despite recent progress (e.g., [14,20,22,28]), the problem of recovering a *spatially dense* 3D representation from many views has not been completely solved. In this document, we describe an approach for automatically decomposing the scene into a set of 3D layers by analyzing the familiar epipolar plane image (EPI) volume. The EPI volume is a dense horizontally rectified spatio-temporal volume that results from a linearly translating camera.

Layers are a powerful way to describe the visual motion of objects in scenes. They capture local coherence, and also make occlusion events explicit. In computer vision, they were first proposed as a method for video compression, where each layer is separately coded and predicted using an affine motion model [31]. A more geometric interpretation was introduced by Baker et al. [2], who combined the idea of layers with a local *plane-plus-parallax* representation.

In computer graphics, the same concept under the name of *sprites* or *layered impostors* was proposed as a means of capturing local appearance and geometry to reuse previously rendered imagery (image-based rendering) [15,24,29]. When combined with the plane-plus-parallax representation, these sprites became *sprites with depth* [23].

Unfortunately, fully automated 3D layer extraction from image sequences has thus far remained an unsolved problem. Torr et al. [30] used a Bayesian approach to perform layer segmentation, but the segmentation was only with respect to a single reference frame, and hence did not capture all of the data contained in the original sequence. Many authors (e.g., [16]) have commented on the large amount of structure inherent in dense motion sequences (4D Lightfields in the most general rigid-scene setting), but as yet there are no algorithms that can successfully analyze this data and break it up into a coherent set of layers.

1.1. Epipolar-plane-image analysis

There are two major parts in this document. In the first part, we develop some novel algorithms to analyze a special kind of spatio-temporal volume (image sequence) to segment it into separate layers. The representation we work with is the epipolar plane image volume (EPI volume) [4]. This volume is constructed by taking a regularly spaced series of images from a camera moving on a linear rail pointing in

a direction perpendicular to the motion (Fig. 1). This volume is equivalent to a simple (orthogonal) 3D slice through the general 4D lightfield of a scene [25]. In their seminal work, Bolles et al. [4] showed how the volume could be analyzed by finding paths and surfaces in this spatio-temporal volume. However, no full (dense) 3D reconstruction was ever demonstrated.

Our ultimate goal is to perform automated layer extraction from arbitrary collections of images. For the purposes of this paper, however, we restrict ourselves to specific camera motions that produce regularly sampled EPI volumes for two main

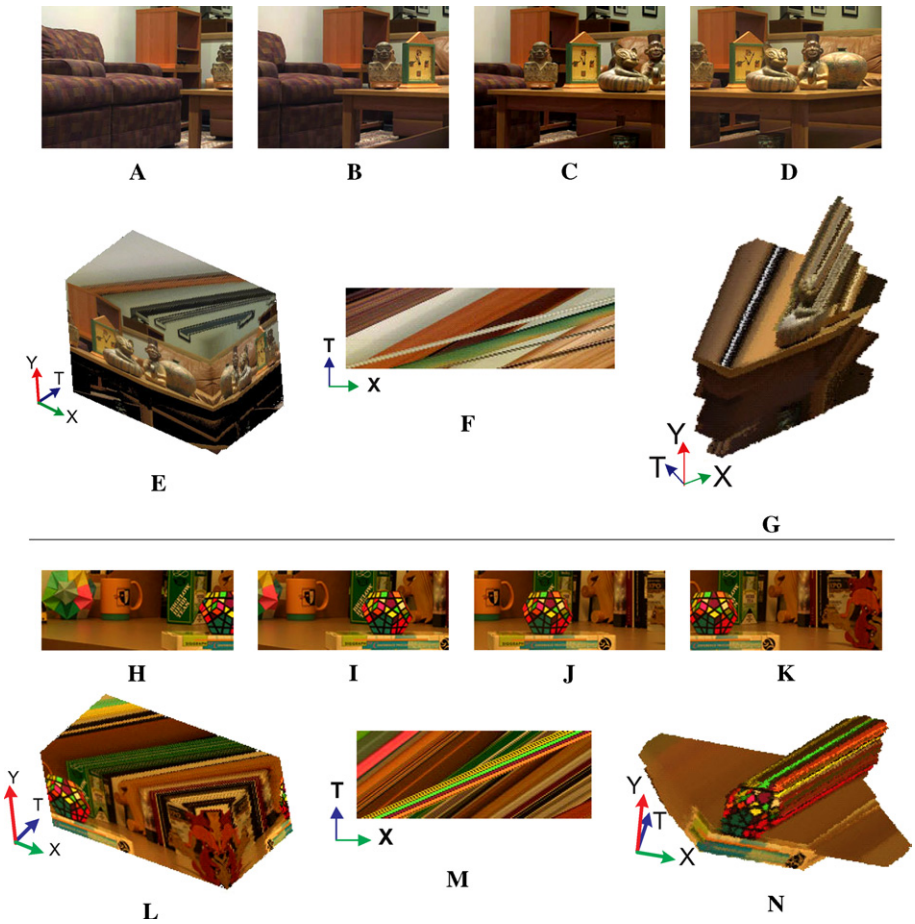


Fig. 1. EPI volumes, strips, and tubes. (A–D) Frames from an indoor sequence. The camera is translating horizontally. (E) The EPI volume corresponding to the indoor sequence. (F) One EPI from that volume. The streaks correspond to different objects in the scenes. The more horizontal the streak, the closer the corresponding object. (G) The automatically extracted EPI-tube corresponding to one of the objects in the scene, namely, the cat statue on the right. (H–K) Frames from another input sequence. (L) The corresponding EPI volume; (M) the EPI corresponding to the 25th scanline in the EPI volume in (L). (N) The automatically extracted EPI-tube corresponding to the dodecahedron in the scene.

reasons. First, the special structure of the frame-to-frame pixel motion makes it much easier to visualize the structure of this data set, and hence to explain the basic algorithms. Second, it admits certain classes of algorithms (such as EPI analysis) that are more difficult to formulate with a general collection of images.

The fundamental new primitives we propose are the *EPI-strip* and the *EPI-tube*.

1.1.1. EPI-strip

An EPI-strip is defined as a quadrilateral on the epipolar plane with two sides aligned with the bottom and top edges of the epipolar plane (corresponding to the first and last frames of the sequence, see Fig. 2).

1.1.2. EPI-tube

A collection of EPI-strips constitutes an *EPI-tube*, i.e., a volumetric primitive with a special ruled surface boundary that represents a coherently moving set of pixels (none of which occlude each other).

An EPI-tube consists, effectively, of a collection of *EPI-trails*, each of which represents the path of a scene point within the EPI volume. When the camera translates linearly and at constant speed, these trails are straight lines in the EPI volume, and their orientation corresponds to the disparity (or inverse depth) of the corresponding scene point.

Figs. 1E and N show the EPI volumes for two of our input datasets. Figs. 1F and O show two EPIs, each associated with a particular horizontal scanline of the corresponding input dataset. Each EPI is automatically decomposed, by the algorithms described in this document, into a set of EPI-strips. An example of automatic EPI-strip extraction can be seen in Fig. 10B. Figs. 1G and P show two EPI-tubes, each segmented out of its respective input EPI volume by our algorithm.

In this document, we describe two algorithms for extracting layers from EPI volumes. The first algorithm extracts single EPI-trails and groups them by analyzing the disparity hypotheses for the trails. The second algorithm extracts whole EPI-strips by directly analyzing collections of EPIs that constitute the EPI volume. Both algorithms operate in two phases. In the first phase, the EPI volume is segmented into a collection of EPI-tubes that account for the appearance of all the pixels. In the second phase, each EPI-tube is described by a simpler layer description, e.g., a single

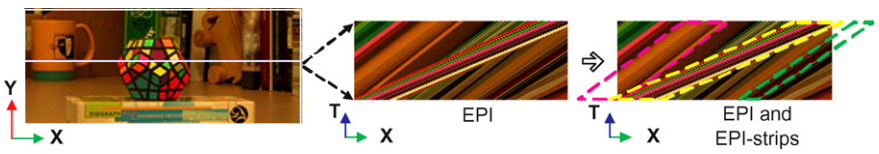


Fig. 2. Definition of an *EPI-strip*. An EPI-strip is a quadrilateral on the epipolar plane with two sides aligned with the bottom and top edges of the epipolar plane. Notice that portions of an *EPI-strip* may be occluded by other EPI-strips or may lie outside the field of view of the available input images. See also Figs. 1F and O.

α-matted image painted onto a 3D plane with optional per-pixel parallax (i.e., a *sprite with depth* [2,23]).[1]

1.2. Characterization of specularities

In the past some work has also been done in using layers to model translucency and reflections (e.g., [27]). In the second part of this paper we extend that work by: (i) defining metrics to distinguish between traces of specular and diffuse features in the EPI and study the factors on which they depend; (ii) showing the limits to which geometry alone can be used to separate the two layers and propose the use of photometric together with geometric constraints; and (iii) building a taxonomy of specular reflections which aids in the design of hybrid algorithms to separate the diffuse and the specular layers of a scene. Finally, we demonstrate the effectiveness of our approach by automatically estimating diffuse and specular components on real scenes with specularities.

The remainder of the paper consists of the following. Section 2 describes the EPI-tube representation and derives a set of constraints associated with them. Section 3 describes the algorithms for layer extraction from EPI volumes. Results on layer extraction and basic manipulation on real data sets are presented in Section 4. Sections 5 and 6 present an analysis of the geometric and photometric characteristics of specular reflections within the EPI framework. An algorithm which implements the separation of a sequence of images into its diffuse and specular components is described in Section 7. Finally, Section 8 summarizes the conclusions of the paper and discusses our plans for future work.

2. EPI-tubes and layers: representations and constraints

In this section we describe some of the fundamental characteristics of EPI-tubes.

Each EPI-tube is a coherent portion of the EPI volume, i.e., the local orientation of the trails within that volume varies smoothly and the trails within the tube do not intersect each other. The intersections between EPI-trails correspond to occlusion events in the EPI volume, i.e., when one point becomes hidden behind another. This also means that EPI-tubes do not necessarily correspond to objects. A self-occluding object may be represented by multiple EPI-tubes.

More precisely, for a camera moving along the X direction with constant speed B, points move only along the horizontal scanline. The x position of a scene point at time t is given by

$$x_t = x_0 + t\ d, \tag{1}$$

[1] Alternative representations of layers such as texture-mapped polyhedral surfaces are also possible, but will not be explored here.

where x_t denotes the x position of the point at time t, x_0 denotes its initial position at time 0, and $d = B\,f/Z$ denotes the *disparity* [20]. Note that the trails of the points close to the camera (larger disparities) are more slanted in the EPI (Fig. 1).

EPI-tubes occlude each other in the usual occlusion-depth ordering. The EPI-tubes corresponding to occluding objects are more slanted (more horizontal) than the ones related to occluded objects. This can be seen in Fig. 1B. Thus, each EPI-tube has one or more *visible regions* or *volumes* where it is not occluded by closer tubes, and zero or more *invisible regions*, where its pixels are occluded by nearer objects.

Under ideal conditions (Lambertian reflection model, constant exposure, no significant foreshortening or aliasing as the camera moves), an EPI-tube can also be described by the color at any of its cross-sections and the rate of motion of each pixel (which corresponds directly to the disparity). This is similar to the *layered sprite* representation proposed in [2], but we do not use a plane-plus-parallax representation for the layer geometry, but rather encode it as a disparity (depth) map.

The initial goal of our algorithm is to label each pixel in our EPI volume with a distinct EPI-tube index, i.e., to perform a complete discrete labeling of the EPI volume. Each EPI-tube can then be interpreted as a layered sprite.

The various observations made above can be summarized by the following set of constraints, which will be exploited by our algorithms:

- Visually distinct and visible image features give rise to visually distinct EPI-trails, i.e., a sharp intensity or color gradient along an epipolar line in the image results in a visible sharp line in the epipolar plane image.
- A homogeneous color region in the image will result in an EPI-tube of homogeneous color.
- For opaque Lambertian surfaces, the color along the EPI-trail is nearly constant.
- Neighboring EPI-trails that belong to the same EPI-tube/strip will have similar orientations (disparities).
- EPI-trail intersections (visible as Y junctions in EPIs) indicate occlusions and hence EPI-strip/tube boundaries.
- No crossings or Y junctions should occur *inside* an EPI-strip/tube.
- More slanted EPI-tubes/strips (corresponding to closer objects and larger disparity) occlude less slanted EPI-tubes/strips (corresponding to objects farther behind and smaller disparity). Exceptions are when non rigid effects, such as specular highlights, occur.

These constraints associated with EPI-tubes and EPI-strips can be used for extracting them from the input data. Our algorithms operate by analyzing all the EPIs of the EPI volume in parallel. Thus, it is worth highlighting constraints associated with corresponding EPI-strips (that belong to the same EPI-tube) from adjacent epipolar planes:

- The neighboring strips belonging to the same tube will have similar colors and disparities (i.e., similar fate).

- Object boundaries are a subset of the tube boundaries. Hence, the continuity of the boundary shape of an object will result in continuity of strip boundaries in adjacent epipolar image planes.

3. Segmenting the input sequence into EPI-tubes

In the next two sections we describe two different algorithms for extracting EPI-tubes from EPI volumes. The first algorithm analyzes the problem in disparity space [3,8,18,33] while the second one directly analyzes the color data contained in the EPI volume. Both algorithms have been applied to multiple data sets and the results are shown in the respective sections.

3.1. Disparity space image processing

One approach to extracting EPI-tubes from the EPI volume would be to look for easily detectable EPI-trails and to merge adjacent trails into tubes. Each EPI-trail corresponds to a particular *disparity hypothesis* (x_0, y_0, d), where (x_0, d) determine the trail's $x_t = f(t; x_0, d)$ coordinate according to (1), and y_0 determines the trail's y coordinate. The set of all such possible hypotheses forms the *disparity space*, which is an old concept dating back to early cooperative stereo correspondence algorithms [8,18]. More recently, the pixel dissimilarity function sampled on a regular (x_0, y_0, d) grid has been called the *disparity space image* (DSI) [3,33]. Finding correspondences then consists of finding the true surfaces hidden in this disparity-space volume.

The EPI volume and the disparity space volume have interesting duality properties (see [4] for some nice illustrations and examples). EPI-trails, which are lines in the EPI volume, are equivalent to points in DSI (Fig. 3). Conversely, a point in an EPI volume, which corresponds to a pixel observed in a particular image, has a linear trail of possible hypotheses associated with it in a DSI.[2] A generalization of this concept to the full space of 3D rays (the 4D Lightfield) is presented in [9,16]. Disparity space can also be defined as an arbitrary collineation of 3-space [7,28] or even a regular 3D grid [22]. This is useful when dealing with an arbitrary collection of images, but we will not need this concept here.

The DSI is built by shearing the EPI volume at a large number of possible disparities and computing the intensity variances along the vertical direction. The sheared EPI volume can be computed as

$$I_s(x_0, y, t, d) = I(x_0 + t\ d, y, t) \tag{2}$$

and its mean and variance can be computed as

[2] Note that the y coordinate is left unchanged when going between the two spaces, so the duality is between the (x, t) and (x_0, d) spaces.

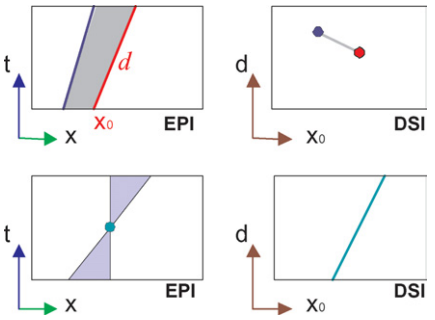


Fig. 3. Duality between epipolar plane image (EPI) and disparity space image (DSI): lines in one space map to points in the other and vice-versa. Top row: The red line in the EPI maps to the red point in the DSI and the blue line in the EPI maps to the blue line in the DSI. Furthermore, the EPI-strip (grey quadrilateral) in the EPI maps to a line segment in the DSI (shown in grey). Bottom row: Dually, the green point in the EPI maps to the green line in the DSI, and the pencil of lines through a voxel in the EPI maps to a line in the DSI. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this paper.)

$$\mu(x_0,y,d)=\frac{\sum_t I_s(x_0,y,t,d)v_s(x_0,y,t,d)}{\sum_t v_s(x_0,y,t,d)} \tag{3}$$

and

$$\sigma^2(x_0,y,d)=\frac{\sum_t [I_s(x_0,y,t,d)-\mu(x_0,y,d)]^2 v_s(x_0,y,t,d)}{\sum_t v_s(x_0,y,t,d)}, \tag{4}$$

where $v_s(x_0,y,t,d)$ is a sheared version of the visibility mask $v(x,y,t)$ that we will define later on (for now, consider it to be 1).[3] Fig. 4 shows some samples of disparity space images $\mu(x_0,y,d)$ and $\sigma(x_0,y,d)$.

Next, we label voxels in the DSI that have a variance lower than the other voxels they might potentially occlude. If the voxel is indeed an occluding voxel, then the occluded voxels should get a photoconsistency measure (variance) that is contaminated by the occluding voxel. Ignoring noise and assuming that the color/intensity of the occluding and occluded voxels are different, we would thus expect the variance associated with an occluding voxel to be smaller than those being occluded.

In our current implementation, we add some slack in the comparison, i.e., we label a voxel (x_k,y_k,d_k) in the DSI as a good candidate of an EPI-tube if for all $(x_j,y_j,d_j)\in S(x_k,y_k,d_k)$ we have $\sigma(x_k,y_k,d_k)<\sigma(x_j,y_j,d_j)+\varepsilon$ (currently, $\varepsilon=1$), where $S(x_k,y_k,d_k)$ is the shadow cast by (x_k,y_k,d_k) in the DSI (Fig. 4D).[4]

[3] To perform the shearing, we shift the original images horizontally by fractional pixel amounts using linear interpolation.

[4] The triangle in the opposite direction from the shadow region is the *free space region* [4], in which no further matches should in principle be allowed. However, we do not currently use this constraints, since errors early on in the matching can preclude valid later matches.

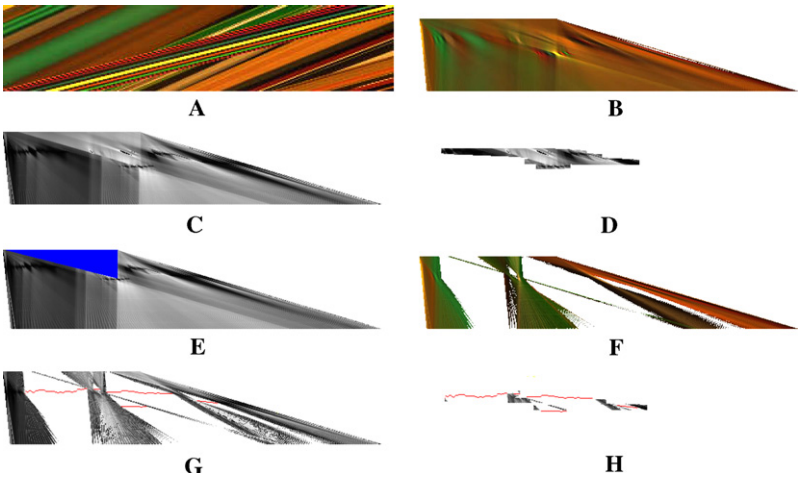


Fig. 4. Sample disparity space images (scanline $y = 45$ of sequence in Figs. 1H–M): (A) EPI $I(x,t)$, (B) mean $\mu(x_0,d)$, (C) standard-deviation $\sigma(x_0,d)$, (D) line-masked std-dev, (F) shadow region (in blue) for one point in the DSI; (F–H) corresponding images after 2nd iteration. The lines in red indicate the extracted EPI-tube components. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this paper.)

Once we have identified a good set of candidate voxels, we find connected regions of such voxels while filling across small gaps. Currently, this is implemented using morphological dilation and erosion operators within the 3D disparity space. Next, we pick a set of layers that do not occlude one another. This is accomplished by casting shadow masks as each layer is being picked from the DSI in front to back order.

Once these layers are chosen, the corresponding voxels in the original EPI volume are then masked out by setting the corresponding $v(x,y,t)$ entries to 0. The DSI is recomputed, and the entire cycle is repeated to extract another set of layers (or to add to current ones). In extracting layers subsequent to the first set, any additional hypothesized layer is tested by reconstructing the EPI with the current set of layers, and computing the error introduced by this additional layer. If the average error introduced exceeds a threshold (currently set at 5 intensity levels), then the hypothesized layer is discarded. Fig. 4 shows the temporal evolution of the DSIs.

The termination condition is that either all the voxels in the EPI volume have been labeled or there are no voxels that satisfy the photoconsistency condition within the threshold. To ensure that all pixels are accounted for, all voxels that were not labeled are assigned the smallest disparity computed for the dataset (another variant would be to set their $d \leftarrow 0$, i.e., to put them on the plane at infinity). A better alternative, which we have not yet implemented, would be to find connected components of unassigned voxels in the EPI volume and to label them with the smallest nearby disparity.

Our algorithm thus shares a lot of ideas with previous voxel carving algorithms [14,22,28]. However, there are several important differences:

- It commits on a group (EPI-tube) basis, not voxel by voxel.
- Each candidate is extracted by comparing its degree of photoconsistency with those it occludes should it be chosen, rather than some absolute threshold.
- Multiple passes are made through the data, extracting the most certain data first, rather than relying on a single threshold for photoconsistency.

Unfortunately, the algorithm described above sometimes suffers the same problem as with voxel carving: it tends to pick the frontmost EPI-tubes that are photoconsistent. This has the effect of breaking up large textureless regions into multiple EPI-tubes.

Our improved algorithm handles this problem by first finding all the strong lines (EPI-trails) in the EPI (Section 3.2). It then masks out areas in DSI that are not near these points or the lines connecting such points (on the theory that surfaces connecting strong features are good candidates for disparity). The last column of Fig. 4 shows such *line-masked* variance images. Shown in red are the current EPI-tube pixels in the DSI.

3.2. Direct extraction of EPI-strips

A second approach to extracting EPI-tubes is to directly analyze the color information contained in the EPI volume.

As noted earlier, for the case of cameras moving linearly at uniform speed, the boundaries of an EPI-strip are straight lines in the epipolar plane. As it is evident from Fig. 2, each EPI can be thought of as a collection of EPI-strips. Each EPI-strip can be parameterized as two lines or four points (the intersections of the two lines with the top and bottom edges of the epipolar plane).

Based on the observations in Section 2, we have designed the following algorithm to automatically extract EPI-tubes (also shown as a block diagram in Fig. 5):

3.2.1. Stage I: EPI-strip hypothesis generation

In this stage, we create a set of EPI-strip hypotheses for each epipolar plane image.

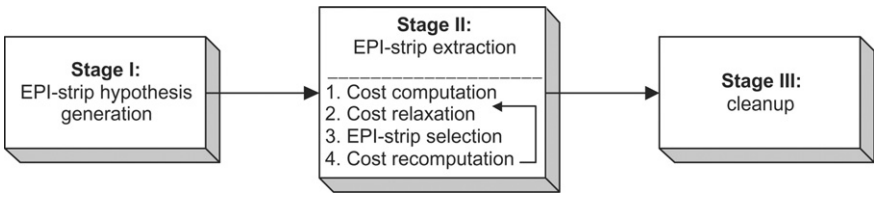


Fig. 5. Block diagram of the direct EPI-strip extraction algorithm.

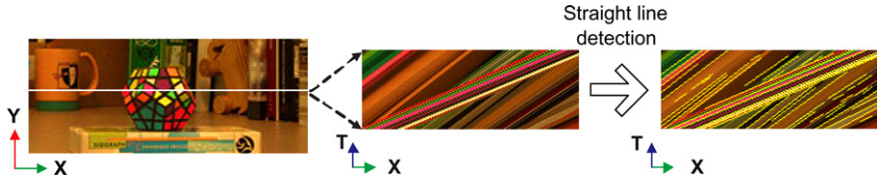


Fig. 6. Candidate EPI-strip boundaries: one EPI with extracted candidate trails superimposed (in yellow). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this paper.)

1. For each EPI extract a set of straight lines corresponding to visible streaks (see Fig. 6, for an example). We use the Canny edge operator to extract edges and then fit straight lines to them. Each line is a potential strip boundary.
2. For each EPI, augment the set of lines with those of its neighboring epipolar planes, by taking the union of the straight lines from the two adjacent planes above and below. This is done in order to overcome possible omissions in Step 1.
3. Sort the straight lines according to their orientation (from most slanted to most vertical, i.e., from highest disparity to lowest).
4. Assuming there are N lines for a given epipolar image plane, create an $N \times N$ upper triangular matrix whose rows and columns are both indexed by the lines. Each element of this matrix correspond to a potential EPI-strip (Fig. 8).

3.2.2. Stage II: EPI-strip extraction

In this stage, we extract EPI-strips separately but simultaneously from each of the EPIs. One EPI-strip is extracted from each EPI at a time using the steps described below. This process is repeated until all the EPI-strips have been extracted from all the EPIs.

1. *Cost computation.* For each EPI, we compute a cost measure for each element of the matrix as follows:
   For each potential strip (a matrix element), rectify (shear) the epipolar plane image so that the selected strip appears vertical in the EPI (Fig. 7). The purpose of this step is to avoid problems due to temporal aliasing and other artifacts.[5] The required geometric transformation is defined by the two boundaries of the strip. The resulting transformation is a geometric bilinear warp, which corresponds to linearly interpolating the disparities of the two boundary pixels (which, in turn, corresponds to assuming a piecewise-planar geometry.)
   Given the rectified EPI-strip, the cost associated with it consists of two parts: (i) a lack of *photo consistency*, which measures the variance (4) of the color information for a pixel across all the views, and (ii) a crossing cost, which penalizes for having non-vertical streaks in the rectified image. This could be because another

[5] Making a strip vertical is equivalent to warping all the input images in order to align the chosen strip over the sequence.

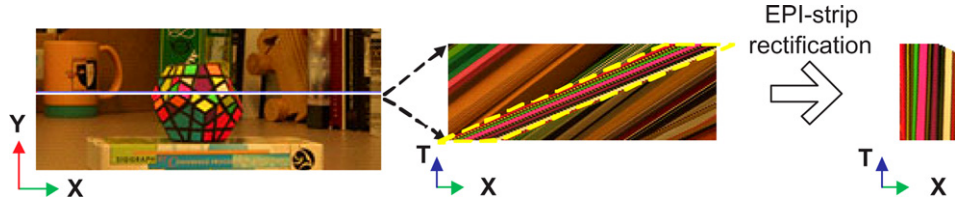


Fig. 7. EPI-strip rectification. The EPI-strip marked in green (dashed line) has been rectified to make its internal streaks vertical. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this paper.)

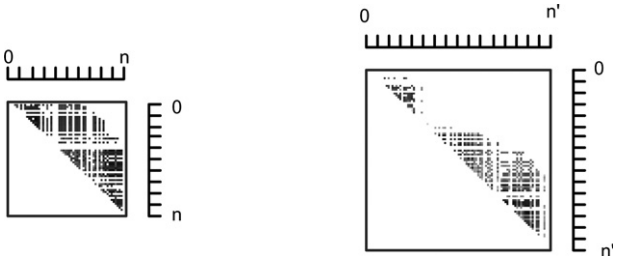


Fig. 8. Cost matrices. Two examples of cost matrices for two different EPIs. The element (i,j) of the matrix indicates the goodness of the EPI-strip defined by the ith and the jth straight line. Notice that the dimensions of the cost matrix varies for each scanline, depending on the number of extracted straight lines.

EPI-strip crosses the current one. The crossing cost is measured in terms of the total squared vertical color gradient (indicating horizontal edges) computed within the hypothesized EPI-strip.

During the first iteration, all the EPI-strips are treated equally. During subsequent iterations (after some strips have been removed), the cost computation is modified as explained in Step 4. Fig. 8 shows the cost matrix associated with a couple of example EPIs. The cost of each EPI-strip is reflected in the brightness of the associated matrix element (low/dark costs are good potential strips).

2. *Cost relaxation.* For a given EPI-strip, consider EPI-strips on neighboring epipolar planes near the given strip. Here, distance is defined in terms of the Euclidean distances between the four corresponding vertices of the EPI-strip quadrilateral. From this set, pick the nearest one to the given EPI-strip and modify the cost of the current strip with a weighted linear combination of the two costs. This is done for all the strips in all the epipolar image planes.
3. *EPI-strip selection.* For each epipolar image plane, traverse the upper-triangular matrix row-by-row from top to bottom (i.e., from most slanted boundaries to the most vertical ones). For each row find the element with the minimum cost in that row. If that element cost is also the minimum for its entire column and is below a predefined admissibility threshold, the element is chosen, and the cor-

responding strip is identified as a valid EPI-strip. This means that the two boundaries of the selected strip are best paired with each other compared to all other possible pairings of either boundary. Also the resulting strip satisfies our aforementioned criteria for being a good strip.

4. *Matrix adjustment and cost recomputation.* After a strip is extracted, the costs of *all* the elements of the matrix are recomputed. This is done by removing the regions of the EPI that are contained within the selected strips from further consideration and marking them as such (Fig. 9). When the cost is recomputed, EPI-strips that completely overlap these regions are set to a maximum cost that results in them being removed from further analysis. The algorithm moves back to the cost-relaxation step above, and the entire process is repeated until no candidate elements are left or none of the remaining ones pass the admissibility cost threshold.

3.2.3. Stage III: Clean up

At the end of Stage II, there may still be regions of an epipolar plane that do not belong to any of the strips that have been extracted. To fill in these "holes," additional EPI-strips are estimated using the following method. Since pixels belonging to an EPI-strip have disparities associated with them, we can detect the pixels in the holes by checking for the lack of any disparity assignment. Each connected component of such pixels is marked a new EPI-strip.

The next task is to assign disparities to the boundaries of these new EPI-strips. There are two possible cases: interior ones that are enclosed by (possibly multiple)

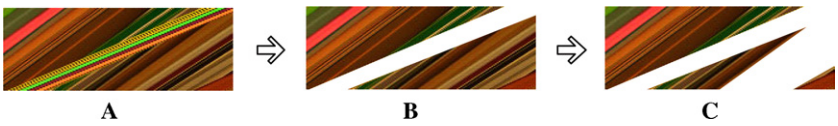


Fig. 9. Strip removal. After extracting an EPI-strip, the corresponding region of the EPI is blanked out and removed from further consideration. (A) Original EPI. (B) The most horizontal EPI-strip (corresponding to front-most object) has been detected and removed from further consideration. (C) The second front-most EPI-strip has been removed. This process of detecting and removing good EPI-strips continues until the current EPI has been completely explained by a set of EPI-strips.

Fig. 10. EPI segmentation into EPI-strips. (A) The EPI corresponding to the 48th scanline of the input sequence in Figs. 1H–M. (B) The EPI has been automatically segmented into different EPI-strips (color-coded, different color for different EPI-strip).

existing EPI-strips, and those adjacent to image borders, which are enclosed by one or more EPI-strips on one side and the image border on the other side. In either case, there may be multiple existing EPI-strips that bound these new strips. For each side of the new strip, we select the bounding strip with the smallest disparity (farthest in the background) as the boundary of the new strip. The disparities of the interior pixels of the new strip are linearly interpolated from these boundaries. For those bordering the image, the entire strip is assigned the same disparity as the boundary on the other side.

Currently, the algorithm produces a separate set of strips for each epipolar plane and computes their disparities. We are working on merging the recovered strips into EPI-tubes.

4. Layer extraction and manipulation

Once the EPI volume has been segmented into a good set of EPI-tubes, we can convert each EPI-tube into a separate *plane-plus-parallax layer* (or *sprite with depth*) [2,23]. To do this, we must first choose a reference frame for each layer. In our current implementation, we simply use the first frame in the sequence. A better choice would be to choose the frame where the majority of the layer surface is best sampled, i.e., where it is most parallel to the image plane.

Once a reference plane has been chosen, we need to compute the per-pixel colors, opacities, and disparities. With our disparity space image analysis algorithm (Section 3.1), this information is already present during the EPI-tube construction. At the end of the DSI analysis, each layer is represented by a collection of DSI voxels. For each voxel, the reference color will have been pulled from the mean image $\mu(x_0, y, d)$ and the disparity is simply the d value. It is then a simple matter to paint these values into the α-matted color (''texture'') image (we use binary opacities for now) and the per-pixel disparity image. For ease of implementation, we currently set the plane equation for each sprite to be the plane at infinity, which means that the inverse depth disparity computed during DSI analysis encodes the correct out-of-plane parallax (after appropriate scaling).

For the EPI-strip extraction algorithm (Section 3.2), since the strips have not yet been merged into tubes, we do not generate sprites. However, we can estimate the per-pixel color and disparity for every strip. The disparity can be easily obtained by linearly interpolating the disparities at the two strip boundaries, and the colors are computed using the median value estimated during the visibility-masked shearing used to compute the original strip cost metric.

4.1. Layer extraction, depth recovery, and object removal

We have run both our algorithms on two different real image sequences. In both cases, the images were acquired by a camera translating sideways (along the scanline direction) moving at a constant speed. We also collected a dense set of viewpoints, keeping the interframe disparities to a few pixels or less.

Results of applying both our algorithms to the input datasets are shown in Figs. 1 and 11. The first row of Fig. 11 shows three frames of the original input sequence. The most visible occlusion event in this sequence is the occlusion of the background by the multi-colored dodecahedron in the foreground.

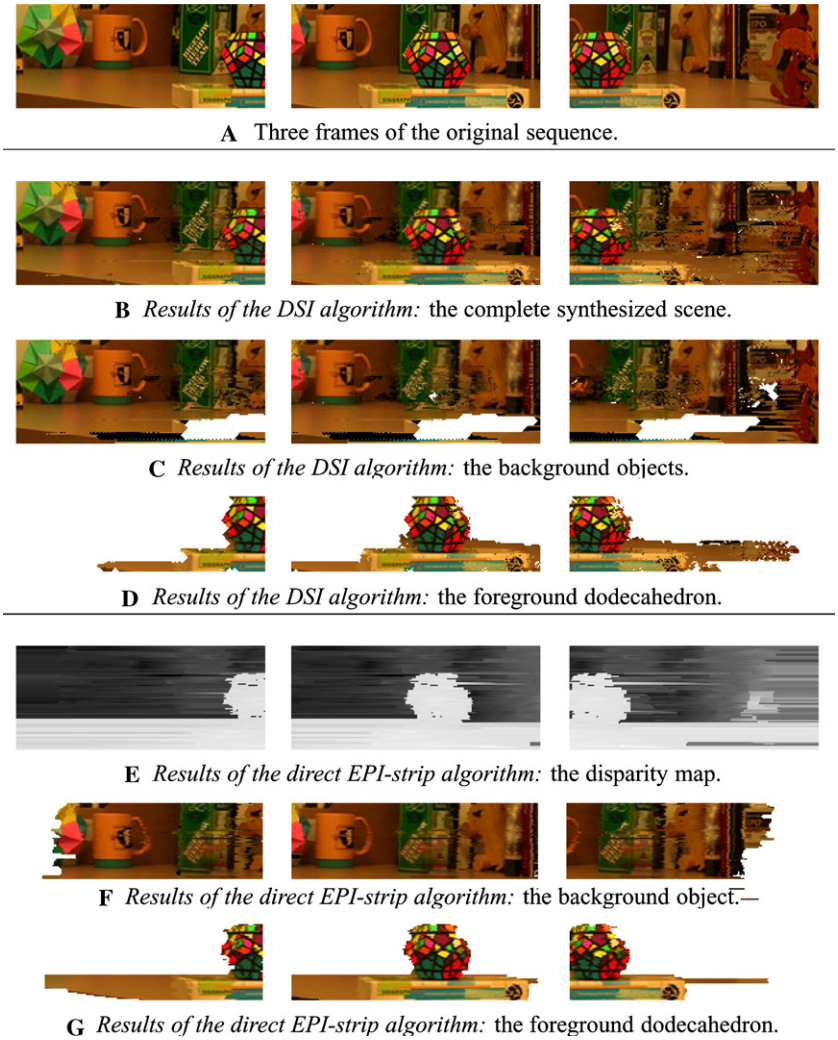


Fig. 11. Experimental results: layer extraction, depth recovery, and object removal. (A) Three frames of the original sequence, (B–D) the corresponding frames of the synthesized sequences obtained using the DSI algorithm, (E) the disparity map for the same scene recovered by the EPI-strip algorithm (corresponding to the three frames in (A)) (F–G) synthesized frames for the background and foreground portions of the scene, respectively, computed using the EPI-strip algorithm.

The DSI algorithm recovered 9 layered sprites (or EPI-tubes) from this sequence. Most of these belonged to the background, while one belonged to the dodecahedron. We used these sprites to re-synthesize the frames of the input sequence. The next three rows (Figs. 11B–D) show the results of the DSI algorithm on this sequence. Fig. 11B shows all the objects in the synthesized scene, whereas Figs. 11B and C show only the objects in the background and foreground, respectively. Note that although the synthesized sequence is noisy in places, on the whole the segmentation and reconstruction is accurate. In particular, note that the boundary of the foreground object is sharp and does not have the fattened edges that are often typical of stereo reconstruction results.

As noted earlier, our current implementation of the EPI-strip extraction algorithm produces EPI-strips for each epipolar plane but does not merge them into tubes. In Fig. 11E we show the disparities estimated by our algorithm. The disparity map looks consistent with the scene layout. Once again, note the sharp discontinuities in the disparity map at the boundary of the dodecahedron in the foreground. Figs. 11F and G show the synthesized frames for the background and foreground, respectively.

Notice that our color estimation process has correctly filled in the areas occluded by the dodecahedron (the green tea box in Fig. 11F). The spatio-temporal layer (EPI-tube) that has been computed for the dodecahedron may be seen in Figs. 11G and 12.

4.2. Object insertion and occlusion handling

The EPI-analysis described above provides us with a complete understanding of the dense 3D geometry of the viewed scene together with a layer-based representation of it. At this point it is quite straightforward to manipulate the extracted EPI-tubes, remove them, duplicate them or insert new objects in the scene, in a coherent 3D fashion. The inserted object may be extracted from a different input video or generated with CAD-like tools. Examples of object insertion are shown in Fig. 13.

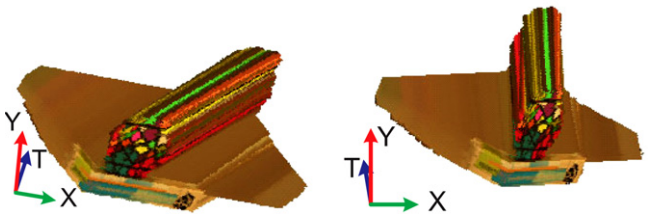


Fig. 12. Extracted EPI-tubes. Two views of the automatically extracted EPI-tube corresponding to the dodecahedron in the input sequence in Figs. 1H–M. Another example, for a different input dataset, may be seen in Fig. 1G.

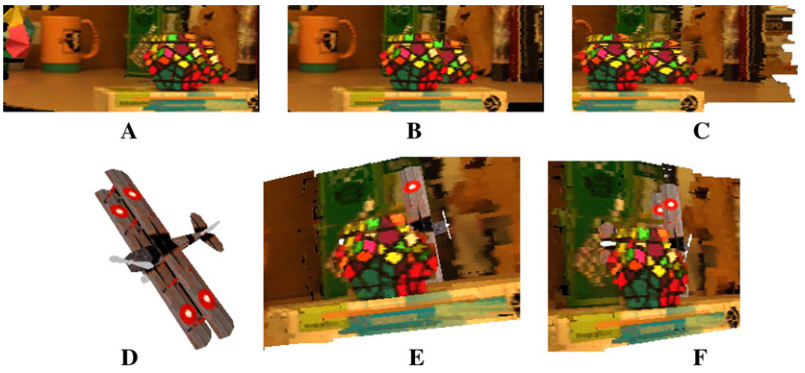


Fig. 13. Object insertion. (A–C) Frames from the augmented "dodecahedron sequence." One more dodecahedron has been inserted behind the original, at a lower level. (D) An image of a 3D model of a toy airplane, to be inserted into the "dodecahedron sequence." (E and F) A complete 3D model of the "dodecahedron sequence" has been obtained from the EPI analysis and the 3D model of the airplane in (D) has been inserted behind the dodecahedron (the front-most layer). Notice that, thanks to the geometric understanding that arises from the automatic EPI segmentation, occlusions are handled correctly.

5. Geometry of specular reflections

In the prior sections, it has been assumed that the scene is Lambertian. In general, however, this hypothesis may be too restrictive. How do we deal with non-rigid effects such as specular reflections? To answer this question, first of all we need to characterize the motion and appearance of specularities.

This section addresses the problem of characterizing the geometric behaviour of specularities and Section 6 deals with their photometric behaviour.

Shiny objects have specular reflections that move in a non-rigid manner when the camera moves. Therefore, specularities must be treated with particular attention. This section presents a mathematical characterization of specular reflections both in terms of their geometry and their appearance within the EPI framework. As we demonstrate in Section 7, this study may be applied to the automatic detection and removal of specular highlights from static scenes. In [17] do obtain some very interesting results on detecting and removing specular highlights from static scenes, but their work lacks the systematic geometric and photometric characterization of specularities addressed in the present work.

Section 5.1 analyzes the simpler case of a 2D reflector, while Section 5.2 deals with the complete 3D case. Section 6 presents the photometric constraints which characterize specular highlights and finally, Section 7 proposes a novel technique for separating diffuse and specular components in static scenes.

5.1. Specular motion in 2D

In general, in the case of flat specular surfaces, the reflected scene point (virtual feature) lies at a single point behind the surface. However, for curved surfaces, the

position of the virtual feature is viewpoint dependent (Fig. 14) [21]. The locus of the virtual feature is a *catacaustic* [10], referred to in this document as just a caustic. Fig. 14A illustrates the caustic curve formed for a circular reflector in 2D and some scene point. Note that any point of the locus of virtual features (caustic) is only visible along the tangent ray to the caustic curve. Also, any two views of a scene containing specularities are insufficient to unambiguously estimate the depth of virtual features (Fig. 14B) necessitating the use of more than two images.

This section analyzes the geometry of reflections for curved surfaces, starting from the simplest case, that of a circular reflector, and then moving to a more general case in Section 5.1.2.

5.1.1. A circular reflector

For purposes of demonstration we assume the specular curve (in 2D) to be circular. The caustic is defined by the geometry of the specular curve and the scene point being reflected. Thus, we can compute the caustic curve in closed form [5,6].

Now, given a camera position, we can derive the point on the caustic where the virtual feature becomes visible. Its image is simply a projection of the caustic point onto the image plane. We derive the image location of a virtual feature as a function of camera pose, specular surface geometry and the scene point.

To compute the EPI trace of the specularities, we assume that the camera motion is linear in the plane parallel to the imaging plane. As stated previously, the linear camera motion implies that the EPI trace of any static scene point must lie along straight lines within the EPI-slice. However, reflected points move along their caustic. Thus, their EPI traces would be expected to be curved.

We define the deviation of an EPI trace from a straight line as disparity deviation (DD). Disparity deviation depends entirely on the movement of the virtual feature and distance of the viewer from the scene. Motion along the caustic in turn depends on the curvature of the surface, surface orientation, and the distance of the reflected point from the surface. The greater this distance, the greater the motion along the caustic surface.

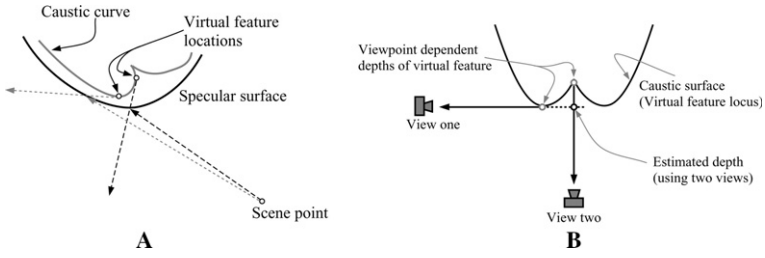


Fig. 14. Reflections on curved surfaces, the 2D case. (A) The geometry of reflection on curved specular surfaces. The position of the virtual feature at two viewpoints lies on the caustic curve at two distinct points. Any point on the caustic is visible only along the tangent to the caustic at that point. (B) Two-view stereo algorithms applied to reflective surfaces would estimate an erroneous depth for the virtual feature, due to lack of sufficient information.

Fig. 15 shows sample EPI curves for two specular curved surfaces. Surprisingly, the curve with higher curvature (Fig. 15B) shows little disparity deviation. In fact, although high curvatures lead to faster angular motion along the caustic, this motion is contained within a very small area. Lower curvatures, on the other hand, can produce noticeable disparity deviation in the EPI. For a given curvature, disparity deviation is accentuated at grazing angles of reflections (as we show below). We now expand the surface to a local cubic approximation, and study the stability of the disparity (trace in the EPI) as a function of curvature and surface orientation.

5.1.2. Infinitesimal motion

Having observed the qualitative behavior of a specularity's trace in the EPI, can we say something more exact about its behavior. In other words, is there a closed form equation that relates local surface curvature, curvature variation, orientation, and the locations of the scene point and camera to the disparity deviation (curvature in the EPI trace)?

Fig. 16 shows a diagram of the 2D case. The scene point being reflected is at S, the camera is at C, and the reflected surface point O is at the origin, with the surface pointing along the x axis. The incident angle to the surface is θ, while the surface itself has a curvature $\kappa = 1/\rho$.

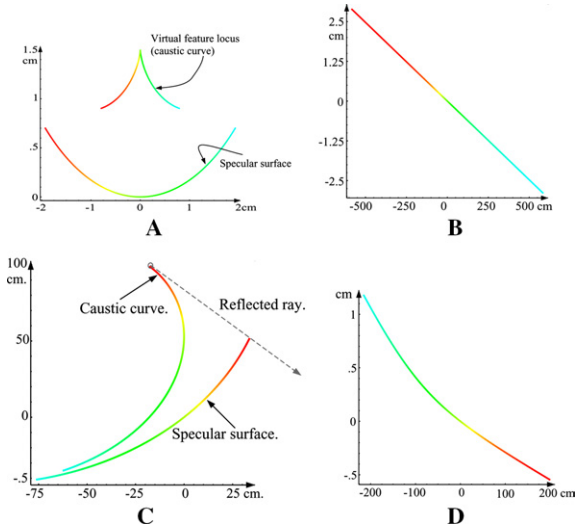


Fig. 15. Plots of specular surfaces, associated caustic curves, and EPI traces. Please note that correspondence between points on the actual surface, caustic curve, and EPI trace, is color-coded. (A) A high curvature surface, such as a soda can, for which the caustic curve is also small and has high curvature. (B) The corresponding EPI trace is almost linear since the virtual feature undergoes minimal motion. (C) An extreme case: the camera observes reflection on an almost flat surface (such as a monitor screen) at an oblique angle. The corresponding caustic has least curvature. Thus for small viewpoint changes, the virtual feature moves significantly. (D) The corresponding EPI trace is noticeably bent having strong *disparity deviations*.

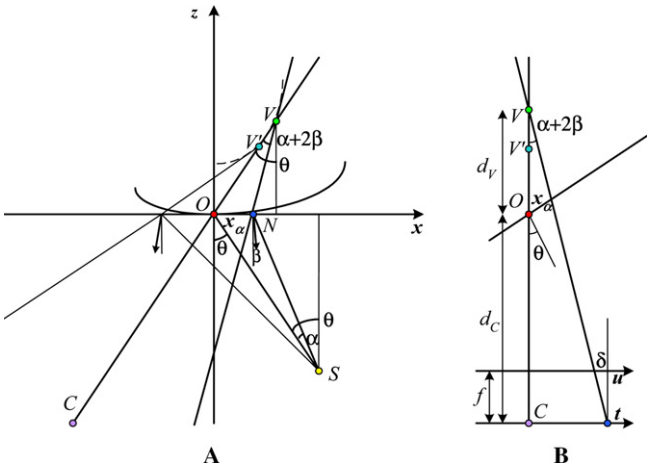


Fig. 16. 2D analysis of specular reflection for generic reflecting geometry. (A) Reflection of point S (source) by the reflecting curved surface at point O as seen by camera C; (B) Projection of the reflected image into the camera C with image plane u moving along the t axis.

Consider an infinitesimal change of angle $\alpha = \angle OSN$ in the direction of the light ray leaving S. This corresponds to a motion along the surface from O to N of length x_α

$$x_\alpha = d_S[\sin\theta - \sin(\theta - \alpha)], \tag{5}$$

where, d_S is the distance from S to O. At the new reflection point N, the surface normal has changed by an angle β

$$\beta = \kappa x_\alpha + \tfrac{1}{2}\dot{\kappa}x_\alpha^2 + O(x_\alpha^3). \tag{6}$$

(Note that we explicitly model the change in surface curvature $\dot{\kappa}$, as this will be important in determining the reflection's stability.) Thus, while the incidence angle is $\theta - \alpha$, the emittance angle is $\theta - \alpha - 2\beta$.

This emittance angle determines the angle $\angle OVN = \alpha + 2\beta$, where V is the virtual image point, formed by the intersection of the reflected ray at the origin and the reflected ray at the new point N. We obtain:

$$x_\alpha = d_V[\sin\theta - \sin(\theta - \alpha - 2\beta)], \tag{7}$$

where, d_V is the distance from V to O.

Equating (5) and (7) we obtain

$$d_V = d_S \frac{\sin\theta - \sin(\theta - \alpha)}{\sin\theta - \sin(\theta - \alpha - 2\beta)}. \tag{8}$$

The limit of the above expression as $\alpha \to 0$ gives us the location of the virtual image V. (Note that if the image is stable, as is the case for a planar reflector $\beta = \kappa = \dot{\kappa} = 0$, $d_V = d_S$ is the same for all values of α.)

Applying L'Hospital's rule to the limit of (8) and simplifying, we get

$$\lim_{\alpha\to 0} d_V = \frac{d_S}{1 + 2d_S\kappa\cos\theta}. \tag{9}$$

How does this virtual depth vary in practice? In the limiting case as $d_S \to \infty$ or $\kappa \to \infty$ ($\rho \to 0$), i.e., as the scene point distance becomes large relative to the radius of curvature, we get

$$d_V = \frac{\rho}{2}\sec\theta. \tag{10}$$

This result is quite intuitive: the virtual image sits at the focal point behind (or in front of) the reflector for head-on viewing condition, and further away for tilted surfaces.

The behavior in the general case when the source is closer to the surface is plotted in Fig. 17A. The virtual depth slowly decreases for a convex reflector as the curvature increases. For a concave reflector, the virtual depth decreases, moving rapidly towards negative infinity as the radius of curvature approaches the object distance (as the object approaches the focal point), and then jumps back to positive virtual depths. The actual distance seen by the camera is $d_V + d_C$, so that impossible apparent depths only occur when $d_V < -d_C$.

These results are consistent with the shapes of the caustics presented previously for the circular reflector. What is more interesting, however, is the stability of the virtual depth as a function of curvature and slant. In other words, as we vary our viewpoint, how does v_D change? The answer to this can be approached by differentiating (8) w.r.t. α and setting $\alpha = 0$, yielding

$$\left.\frac{\partial d_V}{\partial\alpha}\right|_{\alpha=0} = -d_V^2\left(d_S\dot{\kappa}(1+\cos 2\theta) + 4\kappa\sin\theta + 2d_S\kappa^2\sin 2\theta\right). \tag{11}$$

This tells us how the virtual image point V moves as we vary α, e.g., how V moves to V' in Fig. 16A when we replace α with $-\alpha$ (the dashed curve indicates the caustic surface). Note that the first term is due to the change in curvature $\dot{\kappa}$ and becomes negligible for highly slanted surfaces, while the other two terms are due to the surface foreshortening $\sin\theta$ and $\sin 2\theta$.

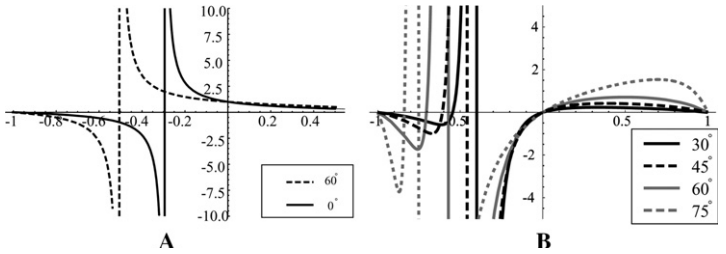


Fig. 17. (A) Plot of virtual depth d_V as a function of curvature κ for $d_S = 1$ and $\theta = 0°$ and 60°. (B) Disparity deviation for $f = 100$ as a function of κ for $d_S = 1$, $d_C = 4$, and $\theta = 30°$, 45°, 60°, and 75°. The horizontal axis in both cases is actually $2/\pi\tan^{-1}\kappa$, so that the full range $\kappa = (-\infty, 0, \infty)$ can be visualized.

Now, how does the disparity (curvature in the EPI) change as we vary the camera position? In other words, what is the disparity deviation of a specular feature? From Fig. 16B, we see that the disparity D is given by

$$D = \frac{\delta}{t} = \frac{f}{d_V + d_C}, \tag{12}$$

which is the usual equation relating disparity to inverse depth.To see how D varies with t, we apply the chain rule to obtain

$$\frac{\partial D}{\partial t} = -\frac{f}{(d_V + d_C)^2} \frac{\partial d_V}{\partial \alpha} \frac{\partial \alpha}{\partial t}. \tag{13}$$

The first partial derivative is given by (11). The second can be computed from the relationship $t = (d_V + d_C)\sin(\alpha + 2\beta)$. For small α and β

$$\begin{aligned} \frac{\partial t}{\partial \alpha} &= (d_V + d_C)\cos(\alpha + 2\beta)\frac{\partial(\alpha + 2\beta)}{\partial \alpha} \approx (d_V + d_C)(1 + 2d_S\kappa\cos\theta) \\ &= (d_V + d_C)\frac{d_S}{d_V}, \end{aligned} \tag{14}$$

using the approximation

$$\beta \approx \kappa x_\alpha \approx \alpha\kappa d_S \cos\theta.$$

Putting all of these together, we get

$$\dot{D} = \frac{\partial D}{\partial t} = \frac{f d_V^3}{(d_V + d_C)^3}\left(\dot{\kappa}(1 + \cos 2\theta) + 4\frac{\kappa}{d_S}\sin\theta + 2\kappa^2 \sin 2\theta\right). \tag{15}$$

Notice that there is no disparity deviation for planar reflection, i.e., $\dot{D} = 0$ when $\kappa = \dot{\kappa} = 0$, as expected.

We can now examine each component in (15). The first ratio $(d_V/(d_C + d_V))$ becomes large when $d_C \approx -d_V$, i.e., when the virtual image appears very close to the camera, which is also when the disparity itself becomes very large. The term that depends on the curvature variation $\dot{\kappa}$ is most significant for frontal surfaces, and decreases for slanted surfaces. It is most significant for undulating surfaces, like the inflection points in a wavy fun-house mirror where things go from "thin" to "fat." At such inflection points, the apparent location of the virtual image can move very rapidly.

The term κ/d_S might at first appear to blow up for $d_S \to 0$, but since d_V is proportional to d_S, this behavior is annihilated. However, for moderate values of d_S, we can get a strong disparity deviation for slanted surfaces. The last term is strongest at a 45° surface slant. It would appear that this term would blow up for large κ, but since d_V is inversely proportionally to κ in these cases, it does not.

To summarize, the two factors that influence the disparity deviation the most are (1) when $d_C + d_V \approx 0$, which is when disparities are very large to start with (because the camera is near the reflector's focal point) and (2) fast undulations in the surface. Ignoring undulations, Fig. 17B shows how $\dot{D}$ varies as a function of κ for a variety of slants, with $d_S = 1$ and $d_C = 4$. Therefore, under many real-world conditions, we ex-

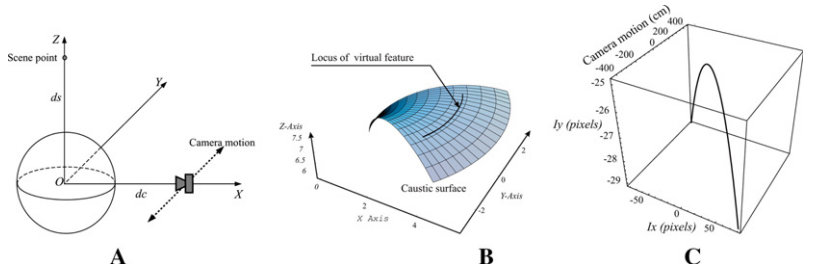


Fig. 18. (A) Analytic setup showing the location of the scene point in relation to the specular surface and camera path. (B) Section of the 3D caustic surface associated with (A). The thin curve on this surface is the locus of virtual features as a function of camera motion. It is clearly seen that the locus of virtual features is neither stationary nor planar. (C) The corresponding EPI-curve clearly exhibits significant epipolar deviations.

pect the disparity deviation to be small enough that treating virtual features as if they were real features should work in practice.

5.2. Specular motion in 3D

We now discuss the effect of specularities in 3D again using the caustic surface to perform our analysis. We present our results for a spherical reflector although the results can be extended to arbitrary surface geometries.

The framework to analyze specularities in 3D is similar to that in 2D. However, in order to simplify the resulting equations, we alter the coordinate frame as well as relative positioning of the scene feature.

Consider a spherical specular surface whose center lies at the origin. The scene point being reflected is located along the positive Z-axis at a distance d_S from the origin. We again derive the caustic surface using the Jacobian technique [6,26]. To study the motion of specularities, we assume the camera to move in the X,Y-plane, parallel to the Y-axis at a distance d_C from the origin[6] (Fig. 18A). Since the reflector surface is symmetric, the caustic is defined by a profile curve in 2D rotated about the Z-axis [26].

We need to derive the image location of a virtual feature as a function of camera pose. We note that the position of the virtual feature locus is essential defined by the caustic surface. Thus, for any camera path, the locus of observed virtual features is a curve in 3D which lies on the caustic surface. For any camera position, the virtual feature not only lies on the caustic surface but is also restricted to the plane defined by the Z-axis and the camera position. The virtual feature therefore is a point on the caustic profile in this plane. Given the camera pose and caustic surface, determining the position of the virtual feature is now reduced to a 2D problem for which an analytic solution exists.

[6] Note, this camera path is not critical to the results we derive.

We know the pose of the camera and hence, can poject any scene point to derive its image location. Thus, the image of the virtual feature is a simple projection of the above derived virtual feature, onto the image plane.

5.2.1. Epipolar deviations for a spherical reflector

Under linear camera motion, the images of a rigid scene point must all lie on the same epipolar line (or plane). However, the motion of a virtual feature on the caustic surface (Fig. 18B) violates this constraint. As seen in Fig. 18C, the image of the virtual feature does not lie on a single scan-line. We refer to this phenomenon as epipolar deviation (ED). The question then arises: how fast does the virtual feature leave any epipolar plane?

In general, epipolar deviations depend on three primary factors: surface curvature, orientation of surface, and distance of the camera from the reflecting surface. We only consider scene points distant from the surface as they usually produce the largest caustic surfaces. We now analyze each factor for its contribution to ED. This study helps determine situations when ED effects can be neglected and when they provide significant cues to the presence of highlights and specularities.

5.2.1.1. Surface curvature. We know that for planar mirrors, the virtual feature is stationary at a single point behind the surface. Similarly, high curvature surfaces such as sharp corners, have very localized tiny caustic surfaces. Between these two extreme curvatures, surfaces exhibit higher epipolar deviations as seen in Fig. 19A.

5.2.1.2. Surface orientation. The angle of incidence of an observed reflection is also critical to epipolar deviation. The more oblique the incidence, the greater the motion of the virtual feature along the caustic surface, causing larger ED. From Fig. 19B we can see how ED drops to zero at an angle which corresponds to the plane in which the caustic curve is planar. Beyond this point, the virtual feature locus is again non-planar and causes epipolar deviations. As one moves to near-normal reflections, we see that the feature locus is restricted to the cusp region of the caustic. This implies very small feature motion, in turn reducing ED.

5.2.1.3. Camera distance. As camera distance from the scene increases, disparity between scene points decreases. Thus, decreasing disparities, imply lower virtual feature motions, in turn decreasing epipolar deviation (Fig. 19C).

To empirically validate these analytical results, we took a series of pictures of some mirrored ball at different distances and orientation, and manually plotted the specularity trajectories (measured to the nearest pixel). As seen in Figs. 19D–F, the results of our experiments are in agreement with our theoretical prediction.

In general, specular reflections or virtual features do not adhere to epipolar geometry. In our geometric analysis, we assume large camera field of view and range of motion, and on occasion, large scene distances. However, in typical real situations, both the camera's range of motion and field of view are limited; as a result, the specular features appear to adhere closely to epipolar constraints. This makes it hard to disambiguate between specular and diffuse EPI-strips solely on the basis of geome-

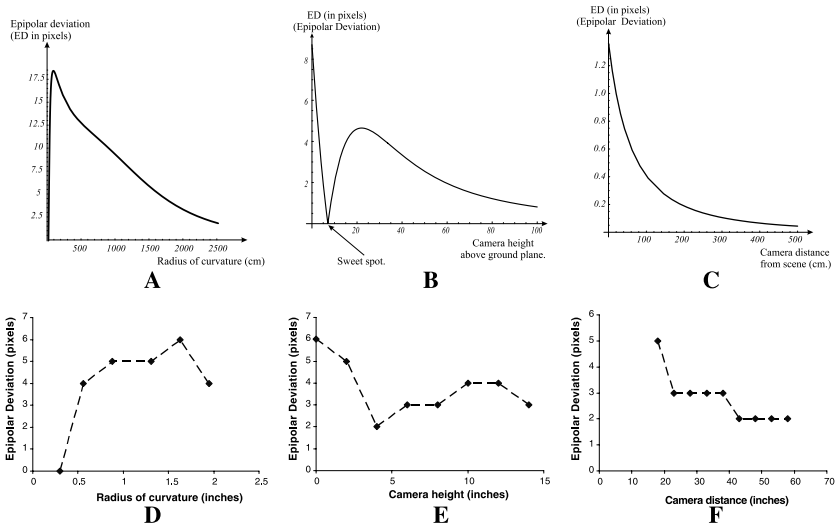


Fig. 19. Epipolar deviations as a function of the three most significant factors. (A) Surface curvature: initially there is a rise in epipolar deviation with respect to increasing radii of curvatures, however, beyond some point ED starts dropping towards zero as the surface flattens. (B) Surface orientation: the epipolar deviation initially dips to zero before rising again and then further reducing back towards zero. This is because, at some surface orientation, the sphere reflects all rays from the scene point in the horizontal plane in which the camera lies. The EPI trace is then restricted to a single epipolar line. (C) Camera distance from scene: this is the most intuitive observation also stemming from stereo parallax, that disparity drops inversely with distance from scene. In the context of virtual feature loci, the dropping disparity reduces effects of the moving virtual feature, in turn reducing epipolar deviation. See the text for more detailed explanations. (D–F) are the corresponding results of experiments using real objects. (D) We used reflective balls with radii ranging from 1.95 to 0.3 in.; each was placed about 3 feet away from the camera. (E) The ball of radius 1.95 in. was placed 3 feet away from the camera. The height of the ball was changed up to 14 in. (F) The same ball was used, with the distance of the camera to the ball varied from 1.5 to 5 feet. Notice the similar trends between the theoretical and experimental plots.

try. As a result, in order for any diffuse-specular separation technique to be effective, photometric characteristics have to be considered as well.

It is clear from the geometric analysis that specular reflections need not adhere to epipolar geometry. Thus, the trace of virtual features across an image under camera motion need not lie in the epipolar plane. Moreover, even if the virtual feature were constrained to the epipolar plane, its trace in the EPI need not be a straight line. In contrast, Lambertian or real scene points always trace out straight lines in the EPI.

Since the traces of specular points (virtual features) can be straight lines as well as curves in the EPI volume, an algorithm that seeks out such "curved tubes" may not be successful. This ambiguity between specular features and Lambertian scene points in the EPI makes geometric constraints necessary but not sufficient. This deficiency of geometry is however, complimented by photometric constraints.

In the next section, we present photometric analysis of specularities under linear camera motion. Results presented in this section motivate the need for hybrid algo-

rithms that use geometric as well as photometric constraints in separating the diffuse layer from the specular layer.

A question that still remains is: "Why do we use EPI analysis to study specularities, if they do not adhere to epipolar geometry?". Given the finite resolution of cameras and sufficient distance of the camera from the scene; epipolar deviation (ED) error is quite limited. The only case in which it is impossible to use EPI analysis when the specular point jumps epipolar/scan-lines between consecutive frames. This in turn corresponds to a large ED. When temporal sampling is high enough and ED low enough (typical scenarios), the specular region lies on a single scan-line long enough to be segmented using photometric and geometric constraints.

6. Photometry of specular reflections

We now present a photometric analysis of specularities under linear camera motion. Within the framework of EPIs, we develop a taxonomy of specularities and motivate the need for hybrid algorithms that use geometric and photometric constraints to separate the diffuse and specular components.

6.1. Taxonomy of specularities

To handle specularities in image sequences, it is instructive to first identify what we consider different kinds of *observed* specularities and their associated photometric behavior. This classification helps in the design and use of "case specific" layer separation algorithms.

We categorize the type of observed specularities based on whether the reflecting and reflected surfaces (which we term *reflector* and *source*, respectively) are textured (Fig. 20). Furthermore, we differentiate between area and point sources, since this has an impact on how separation can be accomplished.

We describe the reflection phenomenon in each of the cases in some detail and explain what separates them from one another. In the analysis to follow we do not assume any attenuation of light as it reaches the viewer. Thus, the result of reflection is simply the addition of light energy such that the reflected component adds to the underlying diffuse component.

6.1.1. Textured reflector—textured source

The EPI-strip associated with this type of specularity is characterized by a blending between the reflector and source textures leading to a criss-cross pattern. With textured surfaces it is difficult to extract individual EPI-strips having the same albedo within the EPI. One has to process each column with the EPI-strip individually. This is equivalent to analyzing every scene point over time.

As the camera moves, a scene point reflects different parts of the surrounding scene. The diffuse component of the surface is bounded by the minimum observed color intensity along any column. In such cases approaches such as those proposed by [27,32] are better suited for separation.

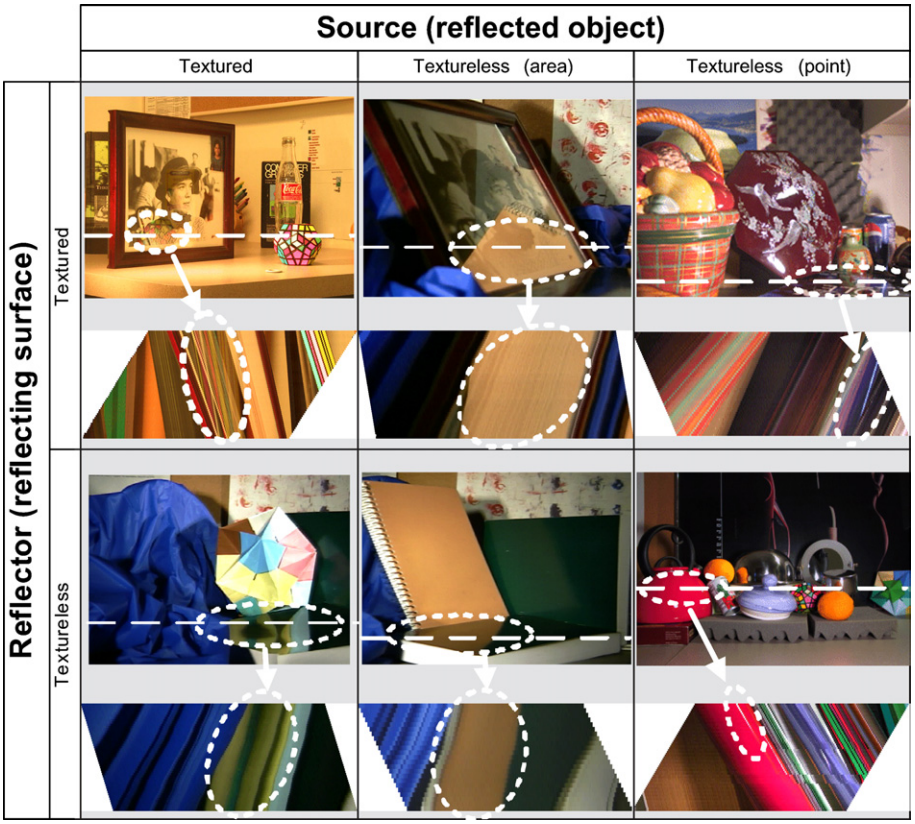


Fig. 20. Taxonomy of specularities with example snapshots of sequences. Below each image is the EPI associated with the marked scan-line. Note that all of the EPIs were sheared for visual clarity.

6.1.2. Textured reflector—textureless area source

In this case, the underlying specular surface is assumed to be textured while the reflected region has almost no texture. Most of the EPI-strip is brightened by a uniform color associated with the source. This may cause ambiguity in separation. We discuss more on EPI ambiguities in the next section.

However, when the EPI-strip is correctly rectified, every column within the EPI-strip corresponds to the effects of reflection of the un-textured source on a single scene point. Each column, when projected in RGB space, would form a dichromatic plane. However, all the columns would form planes which all meet along the same color vector corresponding to the source. Thus, a simple dichromatic model [12] could by used to disambiguate between the two layers.

6.1.3. Textured reflector—textureless point source

In principle, this is similar to the previous case, except that the source is highly localized (Fig. 20). As a result, separation can be accomplished by analyzing con-

stant color sub-strips of the EPI-strip, e.g., using the dichromatic reflectance model [12].

6.1.4. Textureless reflector—textured source

In this case, we assume the underlying specular surface to be textureless. The reflected scene may be richly textured, and the result can be seen in Fig. 20. We also assume that the entire EPI-strip (corresponding to the specular surface) were extracted as a whole. This happens when either there exist multiple scene illuminants or under inter-reflections [13,19]. Extraction of the underlying diffuse component, could then be obtained using a method similar to the multi-chromatic reflection model [1].

6.1.5. Textureless reflector—textureless area source

In this scenario, both the underlying specular surface as well as the reflected scene have no texture. Again, we differentiate between an area being reflected to point reflection. If the reflected region has considerable size with respect to the enclosing surface texture's EPI-strip, ambiguities arise as described earlier. A more detailed explanation of this ambiguity is given in the following section.

6.1.6. Textureless reflector—textureless point source

Once again, this is similar in nature to the above case, except the reflected scene is assumed to be very localized or a point. The difference follows from the fact that this thin trace within the EPI is easier to extract as part of a larger EPI-strip. Thus, aiding its being separated as a specularity (Fig. 20). Possible separation techniques include dichromatic reflectance model [12].

6.2. EPI-strips and their inherent ambiguity

There exists an inherent ambiguity in EPI analysis for specularities and diffuse regions when considering individual EPIs. Fig. 21A illustrates such an EPI. One EPI-

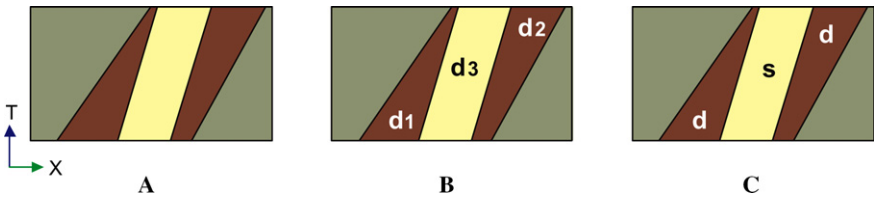


Fig. 21. EPI-strip ambiguity. (A) A typical EPI in which a smaller EPI-strip (dark/brown region) is enclosed with another EPI-strip (light/yellow region). In the absence of prior scene knowledge this EPI may be interpreted in many ways. (B) One interpretation could be that each thin strip of alternating colors, represents a "region" in the scene. Thus, each (d_1, d_2, d_3) is understood to be an unique Lambertian region. (C) Another interpretation could be that the larger EPI-strip (d) includes the lighter EPI-strip (s) within it. This situation can happen only if s is a specularity: geometrically it appears to be behind d, but photometrically it appears to occlude d. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this paper.)

strip (darker) is completely enclosed by another EPI-strip (lighter). Individual layers can now be extracted in several ways leading to valid and unique interpretations.

Fig. 21B is one interpretation where each EPI-strip was extracted separately representing three unique diffuse layers ($d_1, \ldots, d_3$). The varying tilts of their bordering edges in the EPI lead to slanted segments in the scene of varying depths. In contrast, another equally valid extraction includes the inner EPI-strip (Fig. 21C). If this inner strip conforms to the photometric constraints discussed earlier, we interpret it as a specularity s over the otherwise diffuse region d.

Such ambiguities arise in purely Lambertian scenes as well as those containing reflections. In principle, one can reduce the ambiguities by analyzing multiple EPIs all at once. However, this still does not guarantee an ambiguity-free scenario.

6.3. Surface curvature and specularities

As stated in Section 2 within an EPI, closer scene surfaces have a more horizontal orientation than those farther away. For convex surfaces, the locus of virtual features resides behind the surface. Therefore, the corresponding EPI-strip of specular

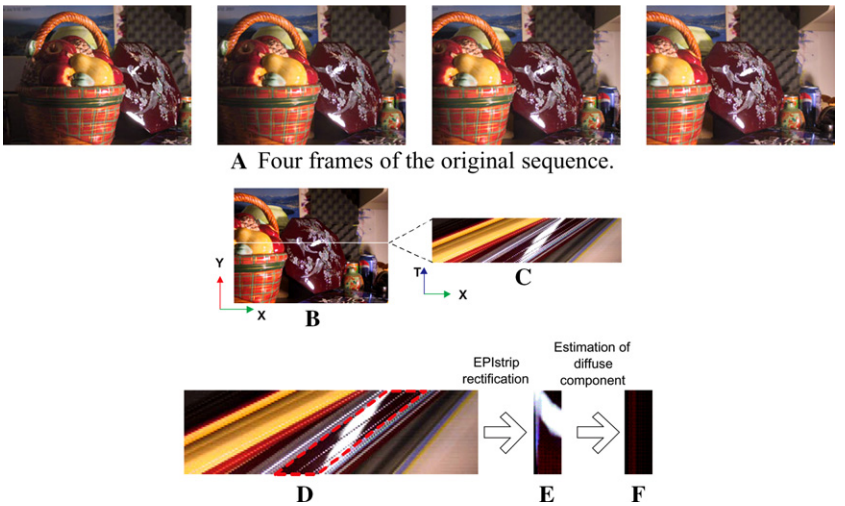


Fig. 22. Estimating diffuse and specular components for each EPI-strip. (A) Some frames from the original input sequence showing some shiny objects, e.g., the central maroon box. (B) One of the input frames with a superimposed scanline. (C) The EPI corresponding to the selected scanline. (D) An EPI-strip selected on the EPI in (C). Notice the typical highlight pattern seen on convex specular surfaces. Chromatically, the highlight region seems to occlude the underlying texture of the surface. However, the orientation of the highlight is more vertical implying a farther depth. This confirms the bright pattern to be caused by a specularity. (E) Rectification of the marked EPI-strip. The diffuse component is now made vertical, while the specular component is oriented beyond 90°. See also Fig. 7 for the rectification of a purely Lambertian EPI-strip. (F) Using photometric analysis along with geometric reasoning, the highlight is extracted and the diffuse component of the selected EPI-strip fully recovered. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this paper.)

reflection has a more vertical orientation than that of the underlying diffuse component (Figs. 22A and B). However, photometrically, this region of the EPI-strip tends to occlude the underlying diffuse component. In contrast, a true occlusion event within an EPI is characterised by the occluding stip having a more horizontal orientation than the underlying surface. Thus, using geometric and photometric techniques, one can dis-ambiguate between occludions and specular reflections.

In contrast, concave surfaces typically form the virtual feature in front of the surface. The EPI-strip of the specular component is therefore more horizontal but restricted to the concave region alone (such as dimples on a regular surface). This is a much harder case to deal with and is beyond the scope of our current work.

7. A technique for removing specular highlights

We now describe a technique for removing the specular components from an image sequence and estimating the underlying diffuse colors associated with the specular regions.

The proposed algorithm first extracts EPI-strips from each EPI (Section 3.2) and then analyzes each individual EPI-strip and decomposes it into its specular and diffuse components. EPI-strips are analyzed for specularities using a variant of [27]. Our technique is more general in that it is designed to work with textured reflectors and all three types of sources shown in the first row of Fig. 20 and is not constrained to planar surfaces.

7.1. Specularity extraction

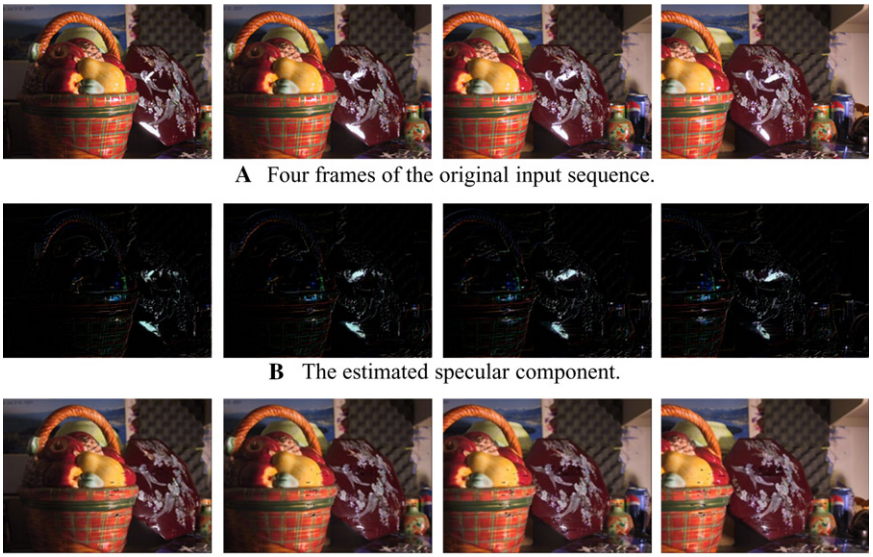
Once the EPI volume has been segmented into a collection of EPI-strips, each EPI-strip is rectified so that trails within it are vertical (Fig. 22).

The scenario assumed here is that of a textured reflector with an arbitrary source. Many highlight regions tend to be saturated in parts. To simplify our process, we look for specularities in EPI-strips containing pixel intensities above a pre-defined minimum value.

In any column of the rectified EPI-strip, the pixel with lowest intensity gives us an upper bound on the diffuse component of that scene point. For every column, we estimate this upper bound and assume the scene point to have the associated color (Fig. 22F). The residual is then the specularity. To validate this step, we group all pixels that are strongly specular and observe their relative orientation within the EPI-strip. If they have a more vertical orientation, then they must be specularities. Note that this is only true for convex surfaces. In our current implementation, we do not consider the effect of concave reflectors.

7.2. Experimental results

To validate our technique, we took an image sequence of a real scene that contains both specular and Lambertian objects. The camera was mounted on a linear



A Four frames of the original input sequence.

B The estimated specular component.

C The estimated diffuse component.

Fig. 23. Automatic separation of diffuse and specular components. (A) A subset of input images of a real scene (already seen in Fig. 22A and repeated here for clarity). (B) The automatically estimated specular component. The specular component is removed from the input sequence thanks to a combined use of geometric and photometric constraints on the behavior of specular highlights. The two strong highlight regions on the maroon box (together with many smaller regions) are correctly detected. (C) The automatically estimated diffuse component. The diffuse component is almost devoid of specular effects. Some artifacts show up in this sequence because of incorrect EPI-strip selection.

translation stage about three feet away from the scene. A set of 50 images was captured at uniform intervals as the camera was translated from left to right. A subset of the acquired images can be seen in Fig. 23A.

This sequence of images were then stacked together to form a spatio-temporal volume on which the EPI segmentation described in Section 3.2 was performed. As seen from Fig. 23B, the specular regions were effectively segmented out from the image sequence. Furthermore, the underlying diffuse component of the scene was recovered successfully in Fig. 23C.

However, inaccurate EPI-strip extraction and interpolation issues while creating the rectified EPI-strip result in some visible artifacts (black spots and residual specularities in Fig. 23C). The same separation result is also shown for a selected scanline in the spatio-temporal volume defined by the input sequence in Fig. 24.

Since we employ a relatively simple technique to detect and separate layers, the results are somewhat sensitive to the EPI-strip segmentation process.

8. Discussion and conclusions

In this document, we have described a new approach for automatically recovering 3D layers from extended multiview sequences by analyzing the data in the entire epi-

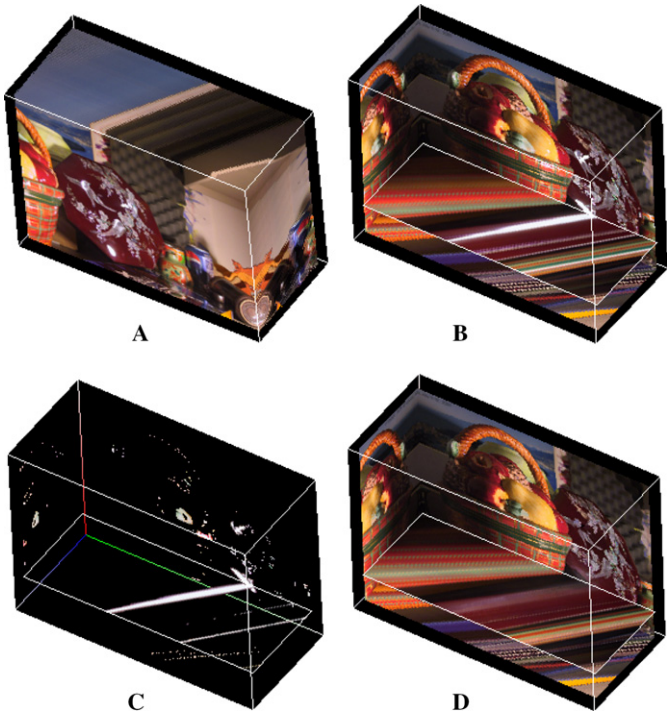


Fig. 24. Specular/diffuse separation in the spatio-temporal volume. (A) The spatio-temporal volume defined by the input sequence in Fig. 22. (B) One EPI from the volume. (C) The detected specularity in the selected EPI. (D) The recovered diffuse component for the selected EPI. The specular streak has been removed and the colour information filled in correctly.

polar plane image volume. Our approach is based on decomposing the EPI volume into a set of EPI-tubes, each of which represents a coherent subvolume corresponding to a coherent portion of the 3D space. The EPI-tubes are the basis for a complete 3D layered sprite representation and for novel techniques to separate diffuse and specular components in static scenes.

We have described two algorithms for extracting EPI-tubes from EPI volumes, and shown their application to real image data sets.

To extend these techniques to non-Lambertian scenes, first of all, we need to characterize the motion and appearance of non-rigid effects such as specular reflections. We performed a geometric analysis of the behavior of specularities in typical scenes, studied their image traces under linear camera motion and introduced the *disparity deviation* and *epipolar deviation* metrics to characterize specular motion. We showed that these deviations depend on surface curvature as well as orientation of the specular surface. There is an expectation that reflections from curved surfaces would always produce curved EPI traces. Surprisingly, both flat and highly curved surfaces do not produce significant deviations. Instead, it is the mildly curved surfaces that

produce the largest deviations. In addition, the closer the object (to the observer), the larger the deviations tend to be.

Such findings point to the possibility of ambiguity in differentiating diffuse from specular components using geometric constraints alone. As a result, geometric analysis must be supplemented with photometric considerations. We have developed a taxonomy of specular reflections to aid in the design of hybrid algorithms that use both geometric and photometric constraints.

Finally, we have presented an application of the EPI analysis for detecting and removing specular highlights from static scenes. Results on real image sequences, using our hybrid algorithm to separate the specular and diffuse component into different layers, show the capabilities of our techniques.

8.1. Future work

Encouraging results have been achieved for specific camera motions (rectilinear with constant velocity in this case) but many of our algorithms extend naturally to the general viewpoint case (e.g., EPI volume shearing is equivalent to a plane-sweep algorithm). In future work, we plan to develop a set of robust algorithms to handle the general viewpoint case. Thus, this work lays the foundations *theory* for multi-image layer extraction, without yet producing an implementation that handles this general case. In the longer term, we would also like to handle dynamic scenes, deforming objects, and the recovery of soft boundaries for layers that better describe the colour mixing that occurs at object boundaries.

Furthermore, we would like to move away from the "local" edge based approach to selecting EPI-strips, to a more global approach. One possibility is that of using "generalized cylinders" to track the contours of image regions across time. This has the advantage of enforcing coherency across scan-lines, while at the same time segmenting the various EPI-tubes. This would of course require a robust image segmentation step.

The final goal of our work is to be able to realistically re-render video sequences from novel viewpoints. To accurately render scenes, we must understand not only their geometry but also their surface properties. By separating the specular component from the diffuse, we can model each independently and achieve better realism as well as high compression rates.

References

[1] R.K. Bajcsy, S.W. Lee, A. Leonardis, Detection of diffuse and specular interface reflections and inter-reflections by color image segmentation, Int. J. Comput. Vision 17 (3) (1996) 241–272.

[2] S. Baker, R. Szeliski, P. Anandan, A layered approach to stereo reconstruction, in: IEEE Comput. Soc. Conf. on Computer Vision and Pattern Recognition (CVPR'98), Santa Barbara, June 1998, pp. 434–441.

[3] A.F. Bobick, S.S. Intille, Large occlusion stereo, Int. J. Comput. Vision 33 (3) (1999) 181–200.

[4] R.C. Bolles, H.H. Baker, D.H. Marimont, Epipolar-plane image analysis: an approach to determining structure from motion, Int. J. Comput. Vision 1 (1987) 7–55.

[5] J.W. Bruce, P.J. Giblin, Curves and Singularities, Cambridge University Press, Cambridge, 1984.
[6] D.G. Burkhard, D.L. Shealy, Flux density for ray propagation in geometrical optics, J. Opt. Soc. Am. 63 (3) (1973) 299–304.
[7] R.T. Collins, A space-sweep approach to true multi-image matching, in: IEEE Comput. Soc. Conf. on Computer Vision and Pattern Recognition (CVPR'96), San Francisco, California, June 1996, pp. 358–363.
[8] P. Dev, Segmentation processes in visual perception: A cooperative neural model. COINS Technical Report 74C-5, University of Massachusetts at Amherst, June 1974.
[9] S.J. Gortler, R. Grzeszczuk, R. Szeliski, M.F. Cohen, The Lumigraph, in: Computer Graphics Proceedings, Annual Conference Series, Proc. SIGGRAPH'96 (New Orleans), August 1996, ACM SIGGRAPH, pp. 43–54.
[10] W.R. Hamilton, Theory of systems of rays, Trans. Royal Irish Acad. 15 (1828) 69–174.
[11] R.I. Hartley, A. Zisserman, Multiple View Geometry, Cambridge University Press, Cambridge, UK, 2000.
[12] G.J. Klinker, S.A. Shafer, T. Kanade, The measurement of highlights in color images, Int. J. Comput. Vision 2 (1) (1988) 7–32.
[13] J.J. Koenderink, A.J. van Doorn, Geometrical modes as a general method to treat diffuse interreflections in radiometry, J. Opt. Soc. Am. 73 (6) (1983) 843–850.
[14] K.N. Kutulakos, S.M. Seitz, A theory of shape by space carving, in: Seventh Internat. Conf. on Computer Vision (ICCV'99), Kerkyra, Greece, September 1999, pp. 307–314.
[15] J. Lengyel, J. Snyder, Rendering with coherent layers, in: Computer Graphics Proceedings, Annual Conference Series, Proc. SIGGRAPH'97 (Los Angeles), August 1997, ACM SIGGRAPH, pp. 233–242.
[16] M. Levoy, P. Hanrahan, Light field rendering, in: Computer Graphics Proceedings, Annual Conference Series, Proc. SIGGRAPH'96 (New Orleans), August 1996. ACM SIGGRAPH, pp. 31–42.
[17] S. Lin, Y. Li, S.B. Kang, X. Tong, H.-Y. Shum, Simultaneous separation and depth recovery of specular reflections, in: Seventh Eur. Conf. on Computer Vision, Copenhagen, Denmark, 2002, part 3, pp. 210–224.
[18] D. Marr, Vision: A Computational Investigation into the Human Representation and Processing of Visual Information, W.H. Freeman, San Francisco, CA, 1982.
[19] S.K. Nayar, K. Ikeuchi, T. Kanade, Surface reflection: physical and geometrical perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 13 (7) (1991) 611–634.
[20] M. Okutomi, T. Kanade, A multiple baseline stereo, IEEE Trans. Pattern Anal. Mach. Intell. 15 (4) (1993) 353–363.
[21] M. Oren, S. Nayar, A theory of specular surface geometry, Int. J. Comput. Vision 24 (2) (1997) 105–124.
[22] S.M. Seitz, C.M. Dyer, Photorealistic scene reconstrcution by voxel coloring, in: IEEE Comput. Soc. Conf. on Computer Vision and Pattern Recognition (CVPR'97), San Juan, Puerto Rico, June 1997, pp. 1067–1073.
[23] J. Shade, S. Gortler, L.-W. He, R. Szeliski, Layered depth images, in: Computer Graphics (SIGGRAPH'98) Proceedings, Orlando, July 1998, ACM SIGGRAPH, pp. 231–242.
[24] J. Shade, D. Lischinski, D. Salesin, T. DeRose, J. Snyder, Hierarchical images caching for accelerated walkthroughs of complex environments, in: Computer Graphics (SIGGRAPH'96) Proceedings, Proc. SIGGRAPH'96 (New Orleans), August 1996, ACM SIGGRAPH, pp. 75–82.
[25] P.P. Sloan, M.F. Cohen, S.J. Gortler, Time critical Lumigraph rendering, in: Symposium on Interactive 3D Graphics, Providence, RI, USA, 1997, pp. 17–23.
[26] R. Swaminathan, M.D. Grossberg, S.K. Nayar, Caustics of catadioptric cameras, in: Proc. Internat. Conf. on Computer Vision, July 2001, pp. II: 2–9.
[27] R. Szeliski, S. Avidan, P. Anandan, Layer extraction from multiple images containing reflections and transparency, in: IEEE Comput. Soc. Conf. on Computer Vision and Pattern Recognition (CVPR'2000), volume 1, Hilton Head Island, June 2000, pp. 246–253.
[28] R. Szeliski, P. Golland, Stereo matching with transparency and matting, Int. J. Comput. Vision, 32(1) (1999) 45–61, Special Issue for Marr Prize papers.

[29] J. Torborg, J.T. Kajiya, Talisman: Commodity realtime 3D graphics for the PC, in: Computer Graphics Proceedings, Annual Conference Series, Proc. SIGGRAPH'96 (New Orleans), August 1996, ACM SIGGRAPH, pp. 353–363.
[30] P.H.S. Torr, R. Szeliski, P. Anandan, An integrated Bayesian approach to layer extraction from image sequences, in: Seventh Internat. Conf. on Computer Vision (ICCV'98), Kerkyra, Greece, September 1999, pp. 983–990.
[31] J.Y.A. Wang, E.H. Adelson, Representing moving images with layers, IEEE Trans. Image Process. 3 (5) (1994) 625–638.
[32] Y. Weiss, Deriving intrinsic images from image sequences, in: Proc. Intl. Conf. on Computer Vision, 2001, pp. II: 68–75.
[33] Y. Yang, A. Yuille, J. Lu, Local, global, and multilevel stereo matching, in: IEEE Comput. Soc. Conf. on Computer Vision and Pattern Recognition (CVPR'93), New York, June 1993, IEEE Computer Society, pp. 274–279.