The Applications of Uncalibrated Occlusion Junctions

A. Broadhurst and R. Cipolla

Department of Engineering, University of Cambridge, Cambridge, CB2 1PZ

{aeb29 │ cipolla}@eng.cam.ac.uk

Abstract

When a scene is viewed from two different viewpoints there are often regions which are only visible in one of the two views. These occluded regions give important information about depth discontinuities in the image. When a background line is obscured by a foreground object, it forms a T-junction in the image, and these junction points can be used to detect occlusion.

Two algorithms are presented in this paper. The first algorithm uses the trifocal tensor to automatically locate T-junctions visible in three views, whilst the second application uses junction points to obtain a constraint for the detection of planar surfaces.

1 Introduction

The stereo correspondence problem is one of the oldest problems in computer vision, yet a general and robust solution remains elusive. One of the major difficulties for stereo matching is the presence of occlusion, although significant advances have been made by computing an occlusion map whilst computing the correspondence [2, 10, 5, 9]. These algorithms allow the smoothing constraint to be suspended near occlusions, however, they only determine occlusion by finding regions where matching under the continuity assumption fails. This often leads to boundaries which are either inaccurate or very dependent on algorithm parameters. Another recent development has been the use of model-based stereo [13], which employs a user supplied model to constrain the matching process. By specifying a model, the user identifies where the occlusions, creases and discontinuities will occur. The matching algorithm can then reliably extract the bas-relief structure of the model.

The major cue to occlusion is the analysis of *binocular junction points* which were introduced to the vision community by Malik [11]. This work is extended, so that junction points can be represented for arbitrary projective cameras, and using this representation, a new constraint, the *trifocal junction point constraint* will be presented. This constraint is used in an algorithm to automatically detecting junction points in images. Finally the concept of an occlusion pseudo-junction will be introduced. These junctions are formed by the intersection of any two lines in the image, and this will lead to a *pseudo-junction planarity constraint* whereby planar surfaces in an image can be detected using only two matched lines.

2   Review

Occlusion occurs when an object closer to the viewer obscures a background object. Since there is a difference in depth between the two objects, the position of the occluded region will be dependent on the camera position. In figure 1 the regions occluded by the foreground object are shaded with two patterns to show in which view they are occluded.

Background object

Foreground object

☒ Occluded in left view
☒ Occluded in right view
☒ Occluded in both views
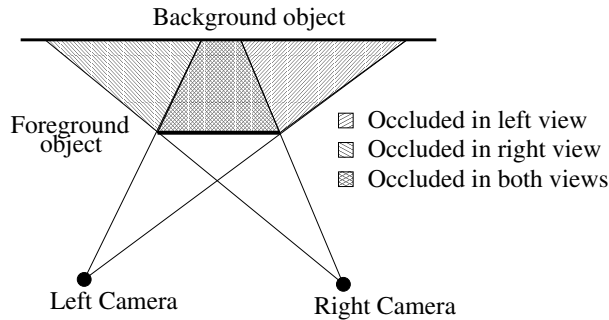
Left Camera

Right Camera

Figure 1: A foreground object obscuring a background causes occlusion.

It will not be possible to match a point on the background object if it is occluded in either of the views. A point which is occluded in only one view is said to cast an *occlusion shadow* into the other view[15]. This corresponds to the diagonally shaded regions in Figure 1, which are also called a half-occluded regions [2].

It has been shown that, in human stereopsis, occlusion plays an important role in the perception of depth discontinuities. These can be seen in the *knife edge* discontinuities of Nakayama and Shimojo [12]. In particular, binocularly viewed T-junction points have been shown by Anderson and Julesz [1] to give important clues to determining whether edges belong to the foreground or background objects.

3   Describing Occlusion Junctions

It was shown by Malik [11] that the angle of the lines forming an occlusion junction are related to the horizontal and vertical disparity. This definition assumes that the images under consideration are rectified, and that the epipolar lines are parallel.

l_β^L y_L l_α^L l_β^R y_R l_α^R

β σ β

α α

x_L x_R

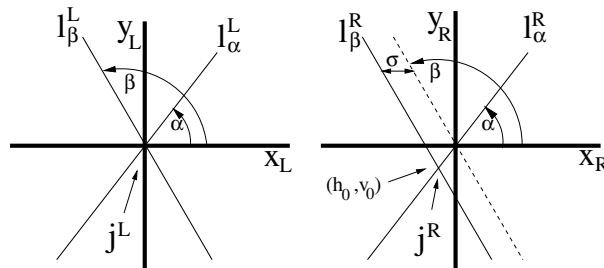(h_0, v_0)

j^L j^R

Figure 2: An occlusion junction (as defined by Malik)

In this paper the effects of occlusion junctions in uncalibrated images will be investigated. Since the epipolar lines are not always parallel, expressing the occlusion junction as a horizontal (along an epipolar line) and a vertical disparity (perpendicular to the epipolar line) is not always meaningful. The occlusion junction will be considered simply as the intersection of the occluding and occluded edges, which are expressed as two straight lines (see Figure 2).

$$\mathbf{j}^L \cong \mathbf{l}_\alpha^L \wedge \mathbf{l}_\beta^L \tag{1}$$

$$\mathbf{j}^R \cong \mathbf{l}_\alpha^R \wedge \mathbf{l}_\beta^R \tag{2}$$

This definition makes no assumption about the nature of the epipolar lines. In the special case of rectified images, with two lines lying on fronto-parallel planes, it can be shown to be equivalent to that used by Malik [11] (See Appendix A).

Since the occlusion junction has now be defined independently of the epipolar geometry, we can use any two cameras related by a fundamental matrix. The only difficulty is that there are no constraints on the motion of a line in two views. Hence both lines that form the junction could move in any direction, so there is no constraint on the occlusion junction in two views. Hence j^L and j^R can be any two points in the two images.

4 Detecting occlusion junctions using three views

It was shown in the previous section that there are no constraints on the motion of the occlusion junction in two views. For this reason it is necessary to consider a third view. The trifocal tensor [8] can be used to project a line from two views into a third view. In particular, the two lines forming the junction can be transfered to the third view :

$$\mathbf{l}_{\alpha i}^3 \cong \mathbf{l}_{\alpha j}^L \mathbf{l}_{\alpha k}^R \mathbf{T}_i^{jk} \tag{3}$$

$$\mathbf{l}_{\beta i}^3 \cong \mathbf{l}_{\beta j}^L \mathbf{l}_{\beta k}^R \mathbf{T}_i^{jk} \tag{4}$$

The occlusion junction will now be given by the intersection $\mathbf{j}^3 = \mathbf{l}_\alpha^3 \wedge \mathbf{l}_\beta^3$. This can be used by an automatic algorithm to detect junction points in the first two images, by hypothesising a match, and projecting it into the third view. The hypothesis can then be verified by checking to see if a junction actually exists at $\mathbf{j}^3$.

4.1 Algorithm

Junction points are detected by segmenting each of the source images, and identifying points where three segments meet. This has proved more reliable than using Canny [4] edges, as the non-maximal suppression breaks junctions. Straight lines were fitted to each of the junction points and all combinations of the junctions in the first and second images were projected into the third view. A junction point was accepted if the point was predicted to within 3.0 pixels, and the angles of the predicted and detected lines agreed to within 10 degrees. Any junction point which could not sustain a straight line for 5 pixels was rejected.

Figure 3: The trifocal junction point constraint. Figures (a) and (b) show straight lines fitted to a junction point. Figure (c) shows the lines projected into the third view.

Figure 4: Automatically matched junction points.

4.2 Results

The junction points which were detected by the algorithm are shown in figure 4. Six points were detected, five of which obey the trifocal junction constraint, and one which is an outlier (CPU case). Of the five correctly matched points only three represent true junction points, as the remaining two represent rigid points in space. These are easily rejected as they obey the epipolar constraint, while the occluding junctions do not.

In a second image sequence of Neville's Court, Trinity College, occluding junctions were detected on the pillars of the archway, and on the rooftop, where the roof occludes the chimneys. These results are shown in figure 5.

Figure 5: Junction points detected using the *trifocal junction point constraint*, from images of Trinity College, Cambridge.

4.3   Discussion

The small number of junctions may appear, at first, to be surprising, however there are two factors which contribute to this result. The requirement that a junction point is visible in all three views, and that it lies on the same edges in all three views, means that most junctions are not matchable. In cluttered areas of the image both edge based and segmentation based algorithms, fail to produce edges of sufficient length to fit accurate straight lines.

The next section of this paper introduces the concept of a pseudo-junction as being the intersection of two matched lines. This enables the junction point constraint to be applied where no explicit junction is visible in all three images.

5   Planarity constraint of two lines

5.1   The occlusion pseudo-junction

In the previous section, it was found that it is difficult to match junction points in three images, as the junction point moves a large distance in the image, for a small change in camera viewpoint. It was also observed, that the lines forming the junction could be matched independently. Hence even if a junction point ceases to be visible in all three views, the location of the junction can still be located by extrapolating the two matched lines which were computed using equations 3 and 4.

5.2   The planarity constraint

Consider two lines in two views that belong to two different surfaces, one foreground and one background. The intersection of these lines will form a occlusion pseudo-junction, and this intersection point will not obey the epipolar constraint. If, however, the intersection does obey the epipolar constraint, then we know that the intersection represents a real point in space. From this we can conclude that the two lines are coplanar.

$$(\mathbf{j}^R)^T F \mathbf{j}^L = 0 \tag{5}$$

$$(\mathbf{l}_\alpha^R \wedge \mathbf{l}_\beta^R)^T F (\mathbf{l}_\alpha^L \wedge \mathbf{l}_\alpha^L) = 0 \tag{6}$$

Equation 5 will only be satisfied if the pseudo-junction point (j^R, j^L) is the intersection of two coplanar lines $(l_\alpha^R, l_\beta^R$ and $l_\alpha^L, l_\beta^L)$. This can be represented as the *junction point planarity constraint* (equation 6).

This constraint will only be reliable if the matched lines are not parallel to the epipolar lines, and that their intersection lies within the image. An alternate proof which does not rely on the use of occlusion junctions is presented in Appendix B.

The conventional way of finding planar surfaces is to use four points, or lines, to compute a homography. This, however, does not guarantee all points will obey the epipolar constraint. A homography (8 degrees of freedom) does not necessarily represent a planar surface (3 degrees of freedom). A further advantage of using two lines is that the number of iterations, required for a RANSAC [7] solution, are significantly reduced as only two lines $O(n^2)$ are required as opposed to four points $O(n^4)$.

5.3 Method

The algorithm considers all pairwise combinations of matched lines. If the intersection of the matched lines lies inside the image, then the intersection point is tested to see if it obeys the epipolar constraint. If so, then these two lines are used to compute the plane normal (3 DOF) for the surface.

Algorithm 1 Algorithm for finding coplanar groups of edges

---

   for each pair of edges **do**
     if the edges intersect within the image **then**
       if the intersection obeys the epipolar constraint **then**
         Find all lines which agree with the initial pair of lines
         Compute the plane equation (4-vector) for the two edges
         Compute the homography for this plane
         Find all the edges which agree with this homography
       end if
     end if
   end for

Each surface normal is computed linearly from the endpoints of the two matched lines, in two views. The two projective camera matrices are $P = [I \,|\, 0]$ and $P' = [A \,|\, \mathbf{e}]$. The rows of matrix A are labelled $\mathbf{a_1}^T$, $\mathbf{a_2}^T$ and $\mathbf{a_3}^T$. The plane is represented by $\Pi = [d_1 \; d_2 \; d_3 \; 1]$, and the image points are given by $\mathbf{p} = [x \; y \; 1]^T$ and $\mathbf{p}' = [x' \; y' \; 1]^T$. The linear solution to the plane equation will now be derived:

$$\text{The plane transfer equation [6]:} \qquad \lambda\mathbf{p}' = (A - \mathbf{e}\mathbf{d}^T)\mathbf{p} \qquad (7)$$

$$\text{using the third row to solve for } \lambda : \qquad \lambda = (\mathbf{a_3}^T - e_3\mathbf{d}^T)\mathbf{p} \qquad (8)$$

$$\text{substitute } \lambda : \quad \mathbf{p}'(\mathbf{a_3}^T - e_3\mathbf{d}^T)\mathbf{p} = (A - \mathbf{e}\mathbf{d}^T)\mathbf{p} \qquad (9)$$

$$\text{rearranging:} \quad (\mathbf{p}'\mathbf{a_3}^T - A)\mathbf{p} = (e_3\mathbf{p}' - \mathbf{e})\mathbf{d}^T\mathbf{p} \qquad (10)$$

$$\text{selecting the first row:} \qquad \frac{(x'\mathbf{a_3}^T - \mathbf{a_1}^T)\mathbf{p}}{e_3\,x' - e_1} = \mathbf{p}^T\mathbf{d} \qquad (11)$$

Equation 11 gives one constraint on the plane vector $\mathbf{d}$. With three or more point matches, a least squares solution can be obtained, for the plane vector $\mathbf{d}$.

5.4 Results

The first experiment, to test the algorithm, uses two images of a calibration grid. Two planes were detected (see figure 6), and the angle between the two planes was found to be 89.91 degrees which is a relative error of 0.1%.

A second sequence of uncalibrated PAL images was captured using a PULNIX camera. The images were projectively calibrated using a set of manual point matches [3], and matched edges were obtained using the automatic edge matching software by Pollard et al. [14]. The results of the algorithm described in Section 5.3 are shown in Figures 7 and 8. Figure 7 shows groups of black lines which agree with planes in the world, and Figure 8 shows the two incorrect planes found by the algorithm.

A second sequence of three images of Fitzwilliam Museum were captured using a digital camera, and the results are shown in figure 9.
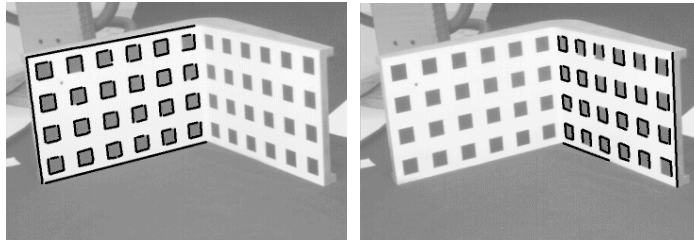
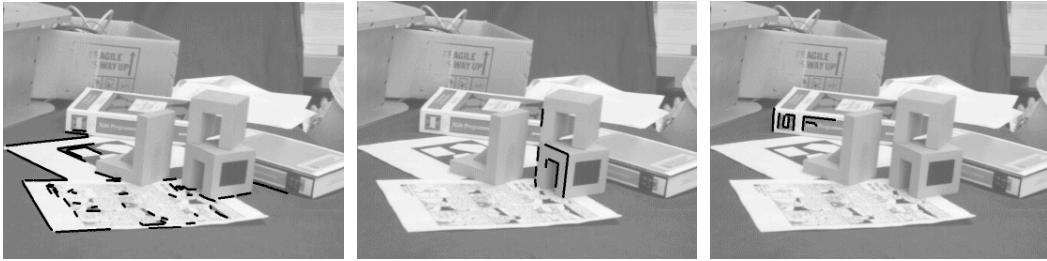Figure 6: Two automatically detected planes of a calibration grid. (angle 89.91 degrees.)

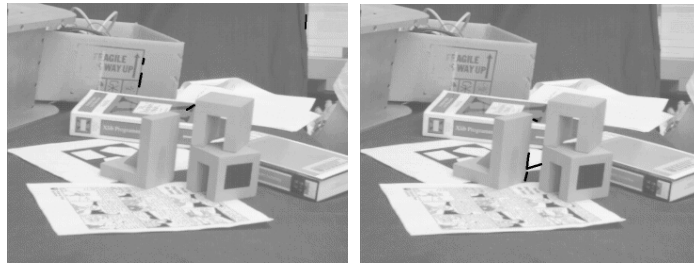Figure 7: The groups of black lines in these three images were found to be coplanar.

Figure 8: This figure shows two groups of black lines which were found by the algorithm to be coplanar, but do not represent real planes in the scene.

Figure 9: The following surfaces of the Fitzwilliam Museum, Cambridge, were found to be coplanar and are displayed using black lines.

5.5 Discussion

To illustrate the importance of this algorithm, the number of plane computations will be considered. In a typical sequence with 400 straight edges, there are 80,000 unordered pairs. After the planarity constraint has been applied, only about 2,000 possibilities remain. It is possible to test each of these possibilities without having to resort to random sampling. In comparison, to perform an exhaustive search of point matches, to compute a homography, would require $400^4 = 2.56 \times 10^{10}$ computations.

6 Conclusion

In this paper two applications of junction points have been investigated. By using a simple representation of a junction point as two lines a trifocal junction point constraint was presented. This was used to develop an algorithm for the automatic detection of junction points in three images. Tracking junction points over many views is not often possible, but it was observed that in many cases, even though the junction point may not be visible, the lines associated with the junction *are visible*. This lead to the concept of an occlusion pseudo-junction, which was the intersection of any two lines in an image. Pseudo-junctions were used to define a planarity constraint for two lines, and an algorithm was demonstrated which used this result to detect planar surfaces in real images.

Acknowledgements

References

[1] B.L. Anderson and B. Julesz. A theoretical analysis of illusory contour formation in stereopsis. *Psychological Review*, 102:705–743, 1995.

[2] P.N. Belhumeur and D. Mumford. A Bayesian treatment of the stereo correspondance problem using half occluded regions. *Proc. Conf. Computer Vision and Pattern Recognition*, pages 506–512, June 1992.

[3] A. Broadhurst and R. Cipolla. Calibration of Image Sequences for Model Visualisation. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume I, pages 100–105, 1999.

[4] J.F. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intell.*, 8:679–698, 1986.

[5] L. Falkenhagen. Block-based Depth estimation from image triples with unrestricted camera setup. *IEEE Workshop on Multimedia Signal Processing*, June 23 – 25 1997.

[6] O. Faugeras. *Three-dimensional computer vision - A geometric viewpoint*. MIT Press ISBN 0-262-06158-9, 1993.

[7] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *CACM*, 24(6):381–395, June 1981.

[8] R.I. Hartley. A linear method for reconstruction from lines and points. *Proc. 5th Int. Conf. on Computer Vision*, pages 882–887, 1995.

[9] S.S. Intille and A.F. Bobick. Disparity-space images and large occlusion stereo. *Proc. 3rd European Conference on Computer Vision*, II:179–186, 1994.

[10] A. Luo and H. Burkhardt. An intensity-based co-operative bidirectional stereo matching with simultaneous detection of discontinuities and occlusion. *Int. Journal of Computer Vision*, 15(3):171–188, July 1995.

[11] J. Malik. On binocularly viewed occlusion juncions. *Proc. 4th European Conference on Computer Vision*, I:167–174, 1996.

[12] K. Nakayama and S. Shimojo. Da Vinci Stereopsis: depth and subjective occluding contours from unpaired images. *Vision Research*, pages 1811–1825, 1990.

[13] C.J. Taylor P.E. Debevec and J. Malik. Modeling and Rendering Architecture from Photographs: A Hybrid Geometry- and Image-Based Approach. *Special Interest Group on Computer Graphics*, 1996.

[14] S. Pollard, M. Pilu, S. Hayes, and A. Lorusso. View Synthesis by Trinocular Edge Matching and Transfer. *Proc. British Machine Vision Conference*, II:770–779, 1998.

[15] Luc Van Gool R. Koch, M. Pollefeys. Multi Viewpoint Stereo from Uncalibrated Video Sequences. *Proc. 5th European Conference on Computer Vision*, II:55–71, 1998.

A Derivation of the junction point motion of two lines

In this appendix the expression for the motion of a junction point from the angles of two lines will be derived using the projective line representation (see figure 10). A junction point is represented as the intersection of two straight lines. This is a more general representation than was used by Malik [11], as it does not rely on rectified cameras. The following proof shows that this new representation can be used to derive Malik's two component disparity (h_0, v_0).
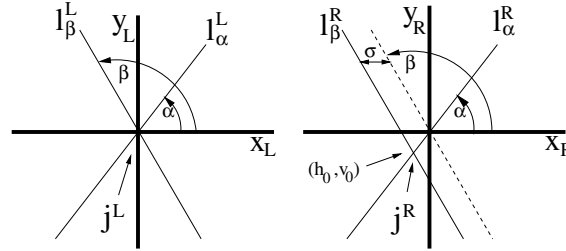

Figure 10: Occlusion junction (as defined by Malik)

$$\mathbf{l}_\alpha^L = \begin{bmatrix} \sin\alpha & -\cos\alpha & 0 \end{bmatrix} \tag{12}$$

$$\mathbf{l}_\beta^L = \begin{bmatrix} \sin\beta & -\cos\beta & 0 \end{bmatrix} \tag{13}$$

$$\mathbf{l}_\alpha^R = \begin{bmatrix} \sin\alpha & -\cos\alpha & 0 \end{bmatrix} \tag{14}$$

$$\mathbf{l}_\beta^R = \begin{bmatrix} \sin\beta & -\cos\beta & \sigma\sin\beta \end{bmatrix} \tag{15}$$

These two lines in the left and right images intersect at $(0; 0)$ and (h_0, v_0) (see figure 10).

$$\mathbf{j}^L = \mathbf{l}_\alpha^L \wedge \mathbf{l}_\beta^L \cong \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \tag{16}$$

$$\mathbf{j}^R = \mathbf{l}_\alpha^R \wedge \mathbf{l}_\beta^R = \begin{bmatrix} -\sigma\cos\alpha\sin\beta \\ -\sigma\sin\alpha\sin\beta \\ \cos\alpha\sin\beta - \sin\alpha\cos\beta \end{bmatrix} \cong \begin{bmatrix} \frac{-\sigma\cos\alpha\sin\beta}{\sin(\alpha-\beta)} \\ \frac{-\sigma\sin\alpha\sin\beta}{\sin(\alpha-\beta)} \\ 1 \end{bmatrix} = \begin{bmatrix} h_0 \\ v_0 \\ 1 \end{bmatrix} \tag{17}$$

This equation has related the angle of the lines forming the occlusion junction to the horizontal disparity, and is the same result as obtained by Malik [11].

B   Alternate proof of planarity constraint.

The matched points ABCD are obtained by intersecting two image lines with two epipolar lines. These four points define a homography H. In addition the point E lying on the intersection of the two lines must also obey the homography H. E will not necessarily obey the epipolar constraint (see figure 11).
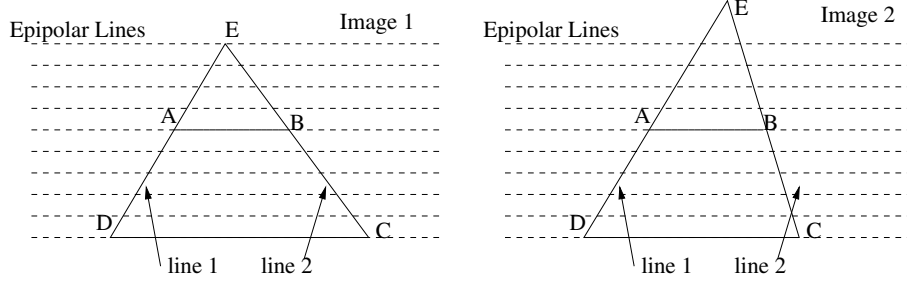

Figure 11: The points ABCD define homography H. Point E will only obey the epipolar constraint if both lines are coplanar.

If the two lines are coplanar they will define a plane $\Pi = [\pi_1 .. \pi_4]$. This plane will induce a homography $\mathbf{x}_R \cong H_\Pi \mathbf{x}_L$ on the image given in terms of the camera matrices $P^L = [I \,|\, 0]$ and $P^R = [A \,|\, \mathbf{a}_4]$ (where $\mathbf{a}_i$ denotes the i-th column of P'). A derivation of this equation can be found in Faugeras [6].

$$H_\Pi = \pi_4 A - \mathbf{a}_4 \begin{bmatrix} \pi_1 & \pi_2 & \pi_3 \end{bmatrix} \tag{18}$$

The epipolar geometry of the two views is described by $x_R^T F x_L = 0$, and the fundamental matrix F can be expressed in terms of the projection matrix $P^R = [A \,|\, \mathbf{a}_4]$ by the equation $F = [\mathbf{a}_4]_\times A$. It will now be shown that homographies of the form (18) must satisfy the equation $\mathbf{x}_L^T H_\Pi^T F \mathbf{x}_L = 0$, which is the epipolar constraint for homography H_Π.

$$
\begin{aligned}
H_\Pi^T F &= (\pi_4 A - \mathbf{a}_4 \begin{bmatrix} \pi_1 & \pi_2 & \pi_3 \end{bmatrix})^T ([\mathbf{a}_4]_\times A) \\
&= \pi_4 \begin{bmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \mathbf{a}_3^T \end{bmatrix} \begin{bmatrix} | & | & | \\ \mathbf{a}_4 \wedge \mathbf{a}_1 & \mathbf{a}_4 \wedge \mathbf{a}_2 & \mathbf{a}_4 \wedge \mathbf{a}_3 \\ | & | & | \end{bmatrix} - \begin{bmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \end{bmatrix} \mathbf{a}_4^T \begin{bmatrix} | & | & | \\ \mathbf{a}_4 \wedge \mathbf{a}_1 & \mathbf{a}_4 \wedge \mathbf{a}_2 & \mathbf{a}_4 \wedge \mathbf{a}_3 \\ | & | & | \end{bmatrix} \\
&= \pi_4 \begin{bmatrix} 0 & \mathbf{a}_1^T(\mathbf{a}_4 \wedge \mathbf{a}_2) & \mathbf{a}_1^T(\mathbf{a}_4 \wedge \mathbf{a}_3) \\ \mathbf{a}_2^T(\mathbf{a}_4 \wedge \mathbf{a}_1) & 0 & \mathbf{a}_2^T(\mathbf{a}_4 \wedge \mathbf{a}_3) \\ \mathbf{a}_3^T(\mathbf{a}_4 \wedge \mathbf{a}_1) & \mathbf{a}_3^T(\mathbf{a}_4 \wedge \mathbf{a}_2) & 0 \end{bmatrix} - \mathbf{0}
\end{aligned}
\tag{19}
$$

$H_\Pi^T F$ is an antisymmetric matrix (equation 19), so $x_L^T H_\Pi^T F x_L = 0$ for all x_L, or in other words, the epipolar constraint is satisfied for all points transfered by homography H_Π, where H_Π is defined in terms of Π.

This argument has shown that in general the intersection of the two lines will not obey the epipolar constraint. In the special case where the lines are coplanar, all points obey the epipolar constraint, and in particular so must point E which is the intersection of the two lines.