A Three-view Matching Algorithm Considering Foreshortening Effects

Ning Xu and Narendra Ahuja
Beckman Institute, University of Illinois at Urbana-Champaign

Abstract

In this paper, we present a three-view matching algorithm that takes into account surface foreshortening effects to match image feature points and provides surface normal directions at these points as a by-product. Compared to other existing, similar algorithms that consider surface foreshortening effects, our algorithm is more efficient since it decomposes a 2D surface orientation search into two 1D searches along two pairs of epipolar lines, and more robust since two pairs of stereo images from three cameras are used and the decomposition makes it possible for our algorithm to detect and match surface points near 3D edges. Experimental results showing correspondences, surface normal directions, reconstructed surfaces are presented and evaluated.

Introduction

The task of matching points between two images is central to computer vision [1, 2]. Many existing stereo matching approaches use a fixed window of *a priori* chosen size [3] or an adaptive window [4] of variable size chosen based on local variations of intensity and disparity. These methods do not take into account viewing geometry and local surface orientation, which will affect the shapes of corresponding windows. Therefore, the accuracy of the above matching algorithms will be affected if this foreshortening effect is considerably large, which is common in wide baseline cases. Devernay and Faugeras [5] propose a fine correlation method that allows a matching window to locally deform between a stereo image pair to estimate accurately both the disparity and its derivatives directly from image data. Maimone [6] models the deformation using local spatial frequency representation, but the scheme still needs brute force search over the surface orientation in addition to the depth search. Hattori and Maki [7] propose an algorithm which computes the surface depth and orientation simultaneously by first getting an initial depth estimate assuming the local surface to be frontal parallel and then considering the surface orientation to refine the depth estimate.

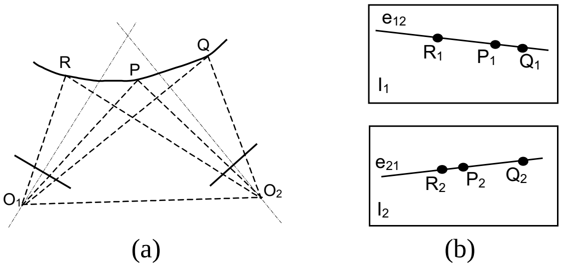
Compared to these methods, our proposed three-view matching algorithm is more efficient and more robust. The use of a three-camera setting to decompose the 2D surface orientation search into two 1D searches along two pairs of epipolar lines makes our matching approach more efficient. These two orientation searches yield an estimate of the surface orientation and a more reliable depth estimate since two pairs of stereo images are used. In addition, our matching algorithm also takes into account the presence of a 3D edge in the vicinity of the point to be matched, so that it can detect and match surface points near 3D edges.

Our Approach

Three calibrated cameras, two displaced horizontally and two vertically, are used to acquire two stereo pairs of images. The distance between each camera and the object is approximately the same. Since their viewing directions are quite different, we do not use rectification as traditional stereo matching algorithms do. Below, we first discuss the matching algorithm for two views and then follow with the three-view cases.

Two views

In Fig. 1(a), O_1 and O_2 are the two camera centers and P is an object surface point. Plane O_1O_2P intersects the object along the surface curve RPQ, whose projection in each image, $R_iP_iQ_i, i = 1, 2,$ is on a straight line, called epipolar line. The two epipolar lines are denoted as e_{12} and e_{21}, respectively, in images I_1 and I_2, as shown in Fig. 1(b). For simplicity, we first consider deformation within the epipolar plane O_1O_2P, which is a one-dimensional deformation along the epipolar line.



(a) (b)

Figure 1: Two camera case: (a) O_1O_2P plane. O_1 and O_2 are two camera centers and P is a point on the object surface. Plane O_1O_2P intersects the object along the surface curve RPQ. (b) Epipolar lines in images I_1 and I_2. Point R_1, P_1, Q_1 and R_2, P_2, Q_2 are images of 3D point R, P, Q on I_1 and I_2, respectively.

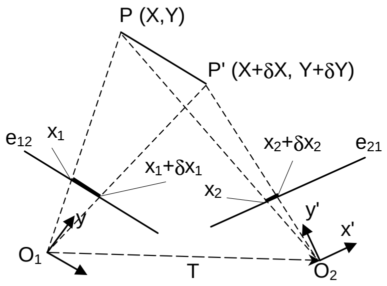


Figure 2: Viewing geometry within the epipolar plane. PP' is a short segment on curve RPQ

The viewing geometry within the plane O_1O_2P is shown in Fig. 2. Suppose the 2D world frame is placed at camera O_1 and the coordinate frame of camera O_2 is related to the world frame with a 2D rigid body motion (R, T). The rotation matrix $R = \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix}$ and the translation vector $T = \begin{bmatrix} T_1 \\ T_2 \end{bmatrix}$ are obtained from calibrated extrinsic parameters. Without loss of generality, we approximate a short curve PP' on the object surface as a straight line where $P = (X, Y)^T$ and $P' = (X + \delta X, Y + \delta Y)^T$. The coordinates of P and P' viewed from camera O_2 are $(X', Y')^T$ and $(X' + \delta X', Y' + \delta Y')^T$, respectively, in O_2's coordinate frame. Camera O_1 projects P and P' onto the 1-dimensional epipolar line e_{12} with local coordinates x_1 and $x_1 + \delta x_1$, while camera O_2 projects them onto the epipolar line e_{21} with local coordinates x_2 and $x_2 + \delta x_2$. From the perspective projection model, we have

$$x_1 = \frac{f_1}{Y} X,$$

$$x_1 + \delta x_1 = \frac{f_1}{Y + \delta Y}(X + \delta X),$$

$$x_2 = \frac{f_2}{Y'}(R_{11}X + R_{12}Y + T_1),$$

$$x_2 + \delta x_2 = \frac{f_2 \cdot [R_{11}(X + \delta X) + R_{12}(Y + \delta Y) + T_1]}{Y' + \delta Y'},$$

where f_1 is the distance between optical center O_1 and epipolar line e_{12}, f_2 is the distance between optical center O_2 and epipolar line e_{21}. Since we are considering a short curve PP', and P' is very close to P, we assume that $\delta Y \ll Y$ and $\delta Y' \ll Y'$. Suppose line PP' is: $Y = aX + b$, then we have $\delta Y = a \cdot \delta X$. From the above equations, we have

$$\delta x_2 = \mathrm{g}(a) \cdot \delta x_1$$

where

$$\mathrm{g}(a) = \frac{f_2}{f_1} \cdot \frac{Y}{Y'} \cdot (R_{11} + a \cdot R_{12})$$

is a function of a, which denotes the orientation of PP' in the epipolar plane. So, if we know the ratio $g(a)$ of δx_2 and δx_1, we can calculate the orientation of line PP' using

$$a = \mathrm{g}^{-1}(\text{ratio}) = \frac{\text{ratio} \cdot f_1 \cdot Y' \cdot R_{12} - f_2 \cdot Y \cdot R_{11}}{f_2 \cdot Y \cdot R_{12}},$$

where Y and Y' follow if the corresponding image points x_1 on e_{12} and x_2 on e_{21} are known, and f_1, f_2, R_{11}, R_{12} follow from the calibration parameters.

By considering the deformation within the epipolar plane O_1O_2P, we can estimate the depth of a given point $P1$ in I_1 and its foreshortening ratio simultaneously using

$$(\hat{d}, \hat{ratio}) = \arg \max_{d, ratio} \text{Corr}(d, ratio),$$

where Corr$(d, ratio)$ is the correlation value based on d and $ratio$. In practice, we use a pair of parallelogram shaped windows with a small height as in Fig. 3 along epipolar lines to correlate point P_1 in I_1 and candidate P_2 in I_2. For each candidate P_2, we change the ratio of the lengths of the two windows. The ratio yielding the best correlation is kept as the match quality of P_2. The candidate yielding the best match quality is selected as the matched point. For the best match P_2, and the corresponding ratio, we estimate both the depth Y, and slope a (and therefore the surface normal at P within the epipolar plane) simultaneously.

To simplify the computation, we make two approximations. First, we do not consider the deformation along the direction perpendicular to the surface curve, which is within plane O_1O_2P. However, from our assumption that $O_1P \simeq O_2P$, this deformation will be very similar for both image I_1 and I_2. Moreover, since we use a parallelogram shaped window with a small height, the deformation differences in the vertical direction can be neglected. Second, the use of rectangular windows aligned with an epipolar line for matching is only an approximation. Truly, the windows should be trapezoid shaped since adjacent rows of the window should lie on adjacent epipolar lines and epipolar lines intersect at epipole. However, in our case, since the epipole is far away from the object in the image, adjacent epipolar lines can be approximated by parallel lines.

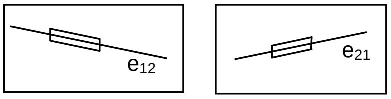


Figure 3: Parallelogram windows with small height along epipolar lines. The width of these windows are varied to yield a best correlation value.

So far, we have not considered the situation when the point of interest P is close to a 3D edge, which is not assured since we select the points based on their image space salience. To address this problem, we consider two different windows extending to the left and right of P and calculate the correlations for each for a range of ratios. If we assume that 3D edges do not occur near P on both sides of it, one of the two windows should yield a match. We distinguish the following cases and draw inferences as indicated:

1. If both sides' match qualities are high, then

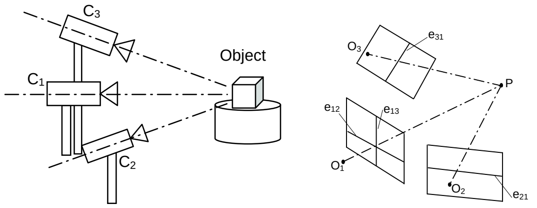
a) if the two ratios are the same, we consider the surface curve as a straight line and its orientation can be estimated from either side's ratio (or both);

b) if the two ratios are different, we consider this point

as a 3D edge point, hence there is no single orientation to be computed rather the two sides have different orientations.
2. If one side's match quality is high and the other's is low, we know that there is a 3D edge point within the window with lower correlation value. The result is still acceptable and the orientation can be estimated using the ratio on the higher correlation side.
3. If both match qualities are low, we regard it as a failed match (e.g., there is an edge point nearby on each side).

Thus, our two-view matching method involves search along the epipolar lines, for a match between two pairs of parallelogram shaped windows with their long edges parallel to the epipolar lines. For each of the two pairs of windows, we vary the ratio of their widths and calculate the correlation values. From the best correlation values and the associated ratios we estimate the depth and slope of the point, and infer whether the point is along a smooth surface curve or there is a 3D edge point nearby along the curve.

Three views



(a) (b)

Figure 4: (a) Three cameras fixating on the object from different viewpoints. (b) Corresponding two pairs of epipolar lines.

Since the above two-view matching algorithm gives the surface orientation only within the epipolar plane, we use three calibrated cameras, C_1, C_2 and C_3 (Fig. 4) and each camera captures one image of the object, yielding three images, I_1, I_2 and I_3.

For each point of interest selected in I_1, a depth range is set based on the assumption that the corresponding point in 3D space is within the fields of view of the other two cameras. For each depth value within this range, we calculate the two corresponding pixels on e_{21} and e_{31}, respectively, and compute the best correlation value while changing the ratios of window widths using the two-view matching algorithm. The correlation values from the two stereo pairs I_1I_2 and I_1I_3 are summed up and the depth associated with the matches with largest summation is selected to be the point's depth estimate. From the stereo pair I_1I_2, we can estimate the orientation of the surface curve within plane PO_1O_2, denoted by $l_1 = (l_{11}, l_{12}, l_{13})^T$, and similarly, we can estimate the orientation of the surface curve within plane PO_1O_3, denoted by $l_2 = (l_{21}, l_{22}, l_{23})^T$. Thus we can estimate the surface normal as $n_p \simeq l_1 \times l_2$.

The depth and surface normal estimates are used to refine the matchings by considering the ordering constraints and the continuity constraints. In our implementation, we simply check the consistency of the ordering and the continuity of depth and orientation of current point with its neighboring points. If a point is inconsistent with the majority of its neighboring points, we assign it a depth corresponding to a local maximum value of the correlation within the acceptable range so that the constraints are satisfied. This step is carried out iteratively until all the points are consistent with the constraints.

Experimental Results

In this section, we present experimental results and evaluate them. In these experiments, the horizontal and vertical camera pairs are about $400mm$ apart, and the object distance is about $500mm$. We calibrate the three cameras using Zhang's *easycalibration* software [8].

The object considered in our experiment is a piece of folded paper with text on it. Fig. 5 shows three input images I_1, I_2 and I_3 from left to right. The marked points in I_1 are the selected points to be matched. We use the three-view stereo matching algorithm to find the corresponding points in the other two images and at the same time estimate the surface normal at these points. The matched points are marked by round dots in I_2 and I_3. In order to check the correctness of the matched points, we manually enter the corresponding points, marked with "x" in I_2 and I_3, and compare them with the points obtained by our algorithm. The average and standard deviation of the distances between the matches found by the algorithm and those marked manually are observed to be 2.0760 pixels and 1.3156 pixels, respectively (in 640×480 images). Please note that this 2-pixel error is not with respect to the ground truth, which is unavailable. Instead, it is the difference between the matches found by our algorithm and the manually clicked matches. Manually clicking itself can easily have a 2-pixel error in a 640×480 image. It is clear that our matching algorithm is consistent with human visual perception.

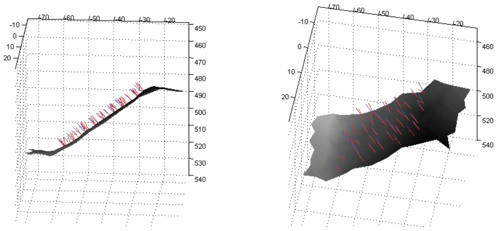


Figure 6: Two different views of surface normal estimates. Red lines show the estimated surface normal direction. Blue lines show the normal directions of least-square fitted planes on neighboring 3×3 points.

The surface normal is estimated as a by-product of our algorithm. Fig. 6 shows two views of a surface along with the estimated normal directions, shown as red lines. To evaluate these estimates, we obtain a least-square planar fit to depths derived from the manually matches in each 3×3 neighborhood away from 3D edges, and use the normal direction of the plane as the ground truth at the neighborhood center. These ground truth normal directions are shown as blue lines. The average and

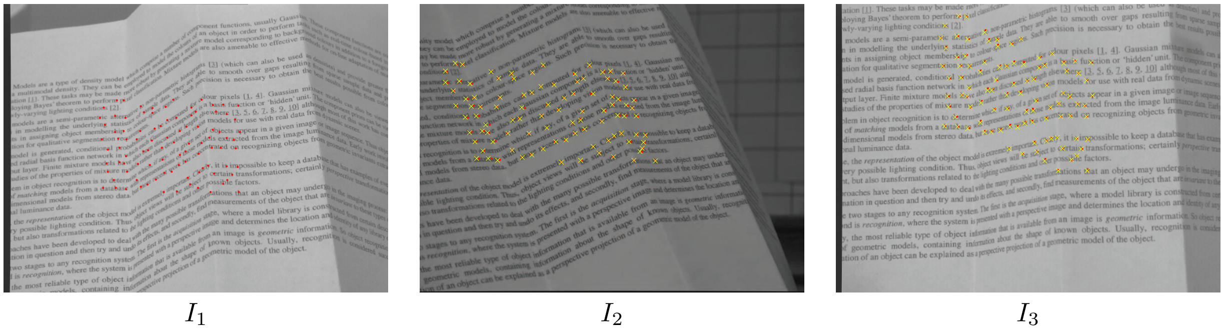


Figure 5: Original images and corresponding points. The red points in I_1 are initial vertices to be matched. Estimated corresponding vertices are shown as red points in I_2 and I_3. The points marked with ”x” are manually supplied matches for evaluation purposes.

standard deviation of the surface normal directions between our estimates and the ground truth are $15.1137°$ and $5.4698°$, respectively. It is reasonably small because when we search for the ratio of one directional deformation, only 10 discrete different ratios are used in each of the two epipolar directions. (If we assume the real surface normal directions are uniformly distributed and there is no error in ratio estimation, the average error of surface normal estimate in each epipolar plane will be $5°$. Moreover, the smallest error in ratio estimation will leads to an average error of $20°$ in that plane.)

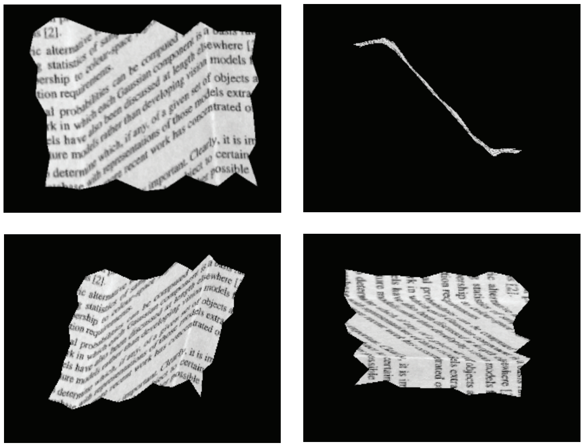


Figure 7: A set of images rendered from novel viewpoints, texture mapped using image I_1. The top right image shows our reconstructed shape is visually accurate.

The 3D model of the surface is reconstructed from the depth values of these points. Fig. 7 shows some rendered images for different viewpoints using the texture from image I_1. As we can see from the top right image in Fig. 7, the reconstructed shape appears to be qualitatively correct.

Summary

Our three-view matching algorithm takes into account the differential foreshortening and achieves robust stereo matching. At the same time, the algorithm estimates surface normal direction as a by-product. Existing surface normal estimation approaches either first obtain the depth and then estimate the normal (thus they are affected significantly by the initial depth estimate), or they involve an exhaustive search for 2D surface orientation in addition to the depth search. Our method estimates depth and surface normal simultaneously. Moreover, it decomposes the 2D orientation search into two 1D searches along two pairs of epipolar lines, and hence is more efficient. The redundancy provided by a second stereo pair adds robustness to the matching. In addition, our matching algorithm also takes into account the presence of a 3D edge in the vicinity of the point to be matched.

References

[1] S.T. Barnard and M.A. Fishler, “Computational stereo,” *Computing Surveys*, vol. 14, no. 4, pp. 554–572, 1982.

[2] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, Cambridge, UK, 2000.

[3] M.J. Hannah, “A system for digital stereo image matching,” *Photogrammetric Engineering and Remote Sensing*, vol. 55, no. 12, pp. 1765–1770, Dec. 1989.

[4] T. Kanade and M. Okutomi, “A stereo matching algorithm with an adaptive window: Theory and experiment,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 9, pp. 920–932, Sept. 1992.

[5] F. Devernay and O. Faugeras, “Computing differential properties of 3d shapes from stereoscopic images without 3d models,” *Technical report 2304, INRIA*, 1994.

[6] M.W. Maimone and S.A. Shafer, “Modeling foreshortening in stereo vision using local spatial frequency,” *Proc. IROS*, pp. 519–524, 1995.

[7] H. Hattori and A. Maki, “Stereo matching with direct surface orientation recovery,” *British Machine Vision Conference*, 1998.

[8] Zhengyou Zhang, “Microsoft easy camera calibration tool,” http://research.microsoft.com/ zhang/calib/.