Method for recovery of heading from motion

Mitsuhiko Hanada and Yoshimichi Ejima

Graduate School of Human and Environmental Studies

Kyoto University

Yoshida-nihonmatsu-cho, Sakyo-ku, Kyoto 606-8501, Japan

Hanada, M., & Ejima, Y. (2000). Method for recovery of heading from motion. Journal of the Optical Society of America A., 17 (6), 966-973

Abstract

Based on Longuet-Higgins & Prazdny's algorithm, a new method was developed. In the algorithm, a radial virtual flow field is generated and the difference between the original velocity field and the virtual radial field is computed. The difference vectors, which are directed to the heading point in the projected plane, allow us to estimate the direction of heading. The simulations of the algorithm were performed and it was shown that the method estimates the direction of heading accurately.

1. Introduction

Recovery of heading from a flow field is an important task for human locomotion such as driving a car or navigating an airplane. It is also important for computer vision. A large number of heading recovery algorithms from image sequences have been presented for computer vision[1-4] and several algorithms were proposed as method in which biological visual systems may recover heading from motion [5-9]. The algorithm of Rieger and Lawton[5] based on the method of Longuet-Higgins & Prazdny [2] is one of the methods which might be used by biological visual systems.

In this paper, we extend the algorithm of Longuet-Higgins & Prazdny[2]. The algorithm is based on the facts that at the location of a discontinuity in depth, there will be a discontinuity in the translation component of the image velocity field, while the rotational component will be roughly constant across the boundary and that if we construct a field of vectors that represent the differences in velocity across the boundaries, these vectors are oriented to the translational field lines.

It is difficult to apply Longuet-Higgins & Prazdny's algorithm to natural scenes because accurate velocity in both sides of an edge is prerequisite to compute the heading direction. Rieger and Lawton extended the algorithm so that it can treat a wide range of motion images [5]. In the algorithm, the difference between each local image velocity and other velocities measured within a restricted neighborhood is computed, and the dominant orientation is computed from the distribution of velocity difference vectors when the distribution is strongly anisotropic, and it is used to compute heading. Hildreth[9] modified the algorithm of Rieger and Lawton further and modeled heading recovery of the biological visual system. Her method works well in situation where noise is large and there are self-moving objects in the scene. This type of algorithm requires a dense-dot field and local depth variations to obtain accurate estimates.

In this study, we present another extension of Longuet-Higgins & Prazdny's algorithm. The extended algorithm has a good characteristic that a dense-dot field and local depth variations are not necessarily required as Rieger & Lawton's algorithm, although the global depth variations are necessary. The algorithm presented here is more effective in some situations than Rieger & Lawton's algorithm. We show simulation results of applying this method to flow fields, to illustrate the effects of the number of sampling points and rotation rates. Finally, we discuss implications for biological model of heading judgment.

2. Algorithm

We assume that an observer (or camera) translates forwardly in the 3-D rigid environment while rotating and the rotation rate is low. We make use of essentially the same notation as Longuet-Higgins and Prazdny[2]. We use a coordinate system that is fixed with respect to an observer, with the Z-axis directed along the optical axis. The X-axis and Y-axis are horizontal and vertical respectively. The translation of the observer in the rigid environment is expressed in terms of translation along three orthogonal directions, which we denote by the vector (U, V, W). U, V and W show translation along the X-axis, Y-axis and Z-axis respectively. The rotation of the observer is expressed in terms of rotation around three orthogonal axes, which we express by the vector (A, B, C). A, B and C, which show rotation around the X-axis, Y-axis and Z-axis, are called pitch, yaw and roll respectively.

We use the equations by Longuet-Higgins and Prazdny[2] to obtain the projected velocity of a point in the 3-D space. The 3-D velocity of a point, P(X,Y, Z) is given by:

$$\begin{aligned} \dot{X} &= -U - BZ + CY \\ \dot{Y} &= -V - CX + AZ \\ \dot{Z} &= -W - AY + BX \end{aligned} \qquad (1)$$

If we consider perspective projection of the velocity onto the image plane, with a focal length of 1 for the projection, the point P on the image (x,y) is given by:

$$\begin{aligned} x &= \frac{X}{Z} \\ y &= \frac{Y}{Z} \end{aligned} \qquad (2)$$

The projected velocity (u,v) in the image plane is given by:

$$u = \frac{(-U + xW)}{Z} - B + Cy + Axy - Bx^2 \qquad (3)$$

$$v = \frac{(-V + yW)}{Z} - Cx + A + Ay^2 - Bxy$$

The first term represents the component of image velocity due to translation of the observer and depends on the depth Z. The remaining terms represent the component of image velocity due to rotation of the observer and do not depend on the depth Z. Here we present a method for recovering the heading from the velocities in the image plane. Because all translation parameters (U, V and W) cannot be recovered by visual information alone, we will present a method for estimating U/W and V/W. The method has five steps.

Step1: Eliminating roll components from a flow field

Step 2: Searching the center of outflow

Step 3: Generating a virtual radial flow pattern

Step 4: Subtracting the virtual radial flow from the original flow

Step 5: Searching the best fitted intersection point of differential flow vectors

Step 1 and 2 are the same as the first part of our previous different method[10]. Step 3 is an original point in this study. Step 4 and 5 were Longuet-Higgins & Prazdny algorithm. In many situations, we can omit Step 1 because roll is generally small when an observer or a camera of a robot moves on the ground.

A. Step1: Eliminating roll components from a flow field

Let (x_i,y_i), (u_i,v_i) and Z_i be the projected position, the velocity and the depth of the i- th sampling point, respectively. We assume that the velocities on a large number of image points are available. We use Hanada & Ejima's method[10] for eliminating roll components from the flow. First we estimate the roll (C) component. When the cloud-like points in the environment are uniformly distributed, C can be estimated by the following:

$$C_e = \frac{1}{N_c} \sum_{\substack{|x_i|>T_{cx} \\ or \\ |y_i|>T_{cy}}} \frac{u_i y_i - v_i x_i}{x_i^2 + y_i^2} \qquad (4)$$

where C_e is the estimation value of C, and N_c is the number of points which satisfy the condition, $|x_i|>T_{cx}$ or $|y_i|>T_{cy}$. T_{cx} and T_{cy} are thresholds. For the case of movement on the ground plane, C is estimated as follows:

$$C_e = \frac{1}{N_{cv}} \sum_{|x_i|>T_{cx}} - \frac{v_i}{x_i} \qquad (5)$$

where N_{cv} is the number of points which satisfy the condition, $|x_i|>T_{cx}$. After the estimation of C, we remove the velocity components of C by redefining v_i as $v_i+C_e x_i$, and u_i as $u_i-C_e y_i$. It is required to know the environment before we apply this procedure. If we use Eq. (4) for a ground plane, some bias in the estimate of C occurs. However, we can use Eq. (5) for a cloud-like environment in order to estimate roll.

B. Step 2: Searching the center of outflow

Let the center of outflow be a point that minimizes the square sum of the distance (d in Fig. 1) between the point and the line passing through the velocity flow vector. Examples of the center of outflow in the ground or cloud-like environment are shown in Fig. 1(b) and (c). Because the low velocities are vulnerable to noise, it may be better to exclude the vectors with low speed for the robust computation of the center of outflow.

Insert Figure 1 about here

C. Step 3: Generating a virtual radial flow pattern

We generate a radial flow pattern. The velocity on the point (x, y) in the image plane is given by:

$$u_{r\,i} = (x_i - x_c) / \tau$$

$$v_{r\,i} = (y_i - y_c) / \tau \qquad (6)$$

where

$$\tau = \frac{1}{N} \sum_{i=0}^{N} \sqrt{\frac{x_i{}^2 + y_i{}^2}{u_i{}^2 + v_i{}^2}} \qquad (7)$$

(x_c, y_c) is the center of outflow. N is the number of sampling points and τ corresponds to the average Z/W of the points[10, 11]. The radial flow is quite similar to the flow pattern generated by simulating translation toward a frontoparallel plane with the average depth while fixating the center of outflow if the rotation rate is low. We will now show that this is true.

We assume that x, y, A and B are small and C is 0. We neglect the quadratic terms about x, y in (3) because they are much smaller than the other terms. Thus we obtain:

$$u = \frac{(-U + xW)}{Z} - B$$

$$v = \frac{(-V + yW)}{Z} + A \qquad (8)$$

We consider a frontoparallel plane with average depth of sampling points. Let P be on the frontoparallel plane and Z_0 be the average depth of all sampling points. The projected image velocity of P is:

$$u = \frac{(-U + xW)}{Z_0} - B$$

$$v = \frac{(-V + yW)}{Z_0} + A \qquad (9)$$

Hanada & Ejima [10] pointed out that one can think that the observer is tracking the center of outflow, whose depth is near the average of other sampling-points' depth when an observer translates and rotates. It means that the center of outflow (x_c, y_c) corresponds to a singular point which has no velocity, on the frontoparallel plane with average depth. Therefore we obtain:

$$\begin{aligned} 0 &= \frac{(-U + x_c W)}{Z_0} - B \\ 0 &= \frac{(-V + y_c W)}{Z_0} + A \end{aligned} \qquad (10)$$

From (9), (10) and $\tau \approx Z_0/W$, we obtain the velocity of a point on the frontoparallel plane as follows:

$$\begin{aligned} u &= \frac{xW}{Z_0} - x_c \frac{W}{Z_0} \approx (x - x_c)/\tau \\ v &= \frac{yW}{Z_0} - y_c \frac{W}{Z_0} \approx (y - y_c)/\tau \end{aligned} \qquad (11)$$

These are essentially the same equations as (6). It indicates that the virtual radial flow is quite similar to the flow pattern generated by translation toward the frontoparallel plane with average depth. The flow field with both the original and virtual flow is sketched in Fig. 2(a). The flow pattern corresponds to the projected motion pattern of the original 3-D points and the points on the virtual frontoparallel plane with the same projected positions as the original image points, as shown in Fig. 2(b).

Insert Figure 2 about here

D. Step 4: Subtracting the virtual radial flow from the original flow

From this step, we use Longuet-Higgins & Prazdny's algorithm[2]. We subtract the virtual radial flow generated in Step 4 from the original flow:

$$\begin{aligned} u_{d\,i} &= u_i - u_{r\,i} \\ v_{d\,i} &= v_i - v_{r\,i} \end{aligned} \qquad (12)$$

E. Step 5: Searching the best fitted intersection point of differential flow vectors

Longuet-Higgins & Paradzny[2] pointed out that if the difference between an image velocity in each point and another velocity measured at the same position with different depth is computed, these vectors are oriented to the heading point in the image. It implies that we get the heading direction by searching a best fitted intersection of the line passing through the differential vectors ($u_{d\,i}$, $v_{d\,i}$). The intersection point corresponds to (U/W, V/W), which indicates the heading direction. Thus, we estimate (U/W, V.W) by computing the center of the differential flow vectors represented by Eq. (12) in the same way as in Step 2.

F. Estimation of rotation parameters except roll.

From Hanada & Ejima[10], we can obtain other rotation parameters than C as follows:

$$A \approx \frac{(\frac{V}{W} - y_c)}{\tau}$$

$$B \approx \frac{-(\frac{U}{W} - x_c)}{\tau} \qquad (12)$$

G. Iterations

We can obtain better estimates by the iterations of the procedure. We remove the rotational components from the flow pattern, then we go back to the step 1 and iterate the procedure. If A or B is large, the estimation may not be good. If we use the iterative procedure, A and B to estimate become smaller as we iterate the process. Therefore we obtain better estimates by the iterations.

The algorithm is very fast because it needs a few iterations and no numerical optimization. The algorithm needs depth variations as Longuet-Higgins & Prazdny's algorithm. In a situation where an observer moves toward a frontoparallel plane, the heading direction cannot be estimated by the algorithm.

3. Simulations

3.A Performance for heading toward a ground plane or random-dot cloud

We performed simulations to test the new algorithm using two environments of a ground and a cloud. They were composed of discrete points whose image motion was determined by translation and rotation of an observer relative to a random dot surface or a cloud in space. 100 trials were conducted for each condition. The motion of the dots on the image plane was computed and these velocities formed the input for heading recovery.

The image subtended 1.05 rad (60 deg) horizontal × 1.05 rad (60 deg) vertical. We used 100 dots. Noise was added to each dot. The velocity of each noise dot was randomly set within 0.01[1/sec] on the image plane and the direction of the noise velocity was random. $|C|$ was less than 0.005 [rad/sec]. Parameter A and B were randomly set to 0.05 [rad/sec] or -0.05 [rad/sec] for each trial. The center of outflow does not correspond to the direction of heading in this situation. The number of iterations was two. In the simulations, we omitted step 1 because $|C|$ was very small. We used all velocities to compute the center of outflow in this simulation for the sake of simplicity.

The following conditions were simulated here;

Ground plane:

In the condition, we focused on translation in the horizontal direction.

*Observer's translation: U was randomly set to a value between –0.125 m/sec and 0.125 m/sec, and W was set between 0.75 m/sec and 1.25 m/sec for each trial. V was 0.

*3-D structure: the observer's simulated eye height was 1.6m from the ground and points covered a plane extending from 2m to 6m in front of the observer. The sight was directed to a point on the ground with depth of 4m.

Cloud plane:

*Observer's translation: U and V were randomly set to a value between –0.125 m/sec and 0.125 m/sec, and W was set between 0.75 m/sec and 1.25 m/sec for each trial.

*3-D structure: Points were placed randomly within a depth range of 2 - 6m.

Figure 3 shows the results of the estimation of the proposed algorithm in the ground condition. The horizontal axis represents simulated heading (U/W) and the vertical axis represents heading estimated by the algorithm. Each point denotes the result of each trial. If the points are scattered along a straight line with slope 1, the estimate is considered to be unbiased. We conducted a linear regression analysis. Deviation from slope of 1 for the regression line shows the bias, and a low correlation coefficient between the regression line and the data points implies variability of data. The slope of the fitting line was 0.91. The direction of heading was statistically underestimated slightly. In other words, the estimate was closer to the direction straight ahead relative to the simulated direction. The square correlation coefficient of the regression analysis was very high (R=.98). The proposed method obtained good estimates of the heading in this ground condition.

Insert Figure 3 about here

Figure 4 shows the results in the cloud condition. We conducted a linear regression analysis. Fig. 4 (a) shows the estimation results of U/W, and (b) shows the results of V/W. The slope of the fitting line was 0.95 in both (a) and (b). The direction of heading was also slightly underestimated in the condition. However, the proposed method obtained good estimates of the heading in this cloud condition.

Insert Figure 4 about here

3.B Effects of the number of dots

We examined the effects of the number of dots. We performed simulations with the smaller number of dots under the cloud condition. The number of dots was 50, 25, 12 or 6. The procedure was the same as in the previous simulations. 100 trials were conducted for each condition. We conducted the regression analyses for the data. The y-intercepts of the regression lines were near 0 in all conditions. In order to evaluate bias and variability, we show slopes of the regression lines and correlation coefficients. They are shown in Fig. 5. The slopes did not change so much across all conditions. The correlation coefficient changed little between 25 to 100 dots, but was smaller for 6 or 12 dots. It indicates that the variability of the estimates was large for 6 and 12 dots. The results of the estimation of the horizontal heading (U/W) and the vertical heading (V/W) were not different. 25 dots were necessary to estimate the heading direction accurately by the algorithm in this condition.

Insert Figure 5 about here

3.C Effects of the rotation rate

We examined the effects of the rotation rate on performance of the algorithm. The number of dots was 100. The absolute values of A and B were 0.1, 0.2, 0.4 or 0.8 [rad/sec]. The signs of A and B were randomly determined for each trial. 100 trials were conducted for each condition. We conducted regression analyses for the data. The y-intercepts of the regression lines were near 0 in all conditions. Slopes and correlation coefficients of the regression lines are shown in Fig.6. The slopes and the correlation coefficients changed little for all conditions. It means that the algorithm maintains the performance up to fairly high pitch and yaw rate (0.8 rad/sec, i.e., 40 deg/sec) although we assume that the rotation rate are low for the algorithm.

Insert Figure 6 about here

4. Discussion

We have presented a new algorithm based on Longuet-Higgins & Prazdny's algorithm[2] . We have performed simulations of the algorithm and have found that the algorithm can achieve good performance. The proposed algorithm is a simple way of heading recovery and is very fast. The algorithm is very useful in situations where fast calculation is necessary.

Longuet-Higgins & Prazdny's algorithm was also extended by Rieger & Lawton[5]. It is needed for Rieger & Lawton's algorithm to select an appropriate neighboring size in order to compute the difference vector. It is a fairly difficult problem because the appropriate size depends on dot density and variation of depth. On the other hand, our method is not confronted with the problem because it uses a virtual radial flow, not velocities in the neighborhood to calculate the difference vectors. In addition, Rieger & Lawton's algorithm requires dense dots, but our algorithm does not necessarily require them though our algorithm also needs a fairly large number of dots (≥ 25 dots). There are some situations where our algorithm is more effective than Rieger & Lawton's. When there are a few dense regions, Rieger & Lawton algorithm works well, but does not work when there are not so many sampling points and they are uniformly distributed in a large-field image because there is no dense part in the image. On the other hand, the algorithm presented in this study works well for that stimulus condition because it does not need dense dot fields. Perrone & Stone used split double frontoparallel planes for their psychophysical experiment as shown in Fig. 7 [12]. When translation toward the double planes was simulated, humans can estimate heading from the stimulus. However, Rieger & Lawton algorithm cannot estimate heading well in this situation because the depth discontinuity between the frontoparallel planes was masked by the central band between the planes and there is no depth continuity in the projected image of each plane. On the other hand, the algorithm in this paper can make good estimation for the stimulus. There exist two focuses of flow for each plane in the stimulus as shown in Fig 7(a) and the center of outflow located between the two focuses. When the virtual radial flow was generated, the flow pattern can be regarded as the flow generated by a frontoparallel plane with the average depth of sampling points as shown in Fig. 7(b). Therefore we can calculate efficient differential velocities and estimate the heading direction. We performed simulations and confirmed that the algorithm recovers heading well for split double planes.

Insert Figure 7 about here

It should be mentioned that there are some limitations of the new method. Firstly, the method can be used for forward translation, not for lateral translation. Secondly, we must know the environmental configuration (cloud or ground) in advance for the accurate estimation of roll. We performed some simulations and found that when we use Eq. (4) for the ground plane, performance of the algorithm worsens slightly although that the effects are not so large. In addition, in order to estimate roll, it is required that the points in the environment are uniformly distributed. Thirdly, when there are the self-moving objects, estimations of the algorithm worsen because rigidity is assumed. Hildreth modified Rieger & Lawton algorithm to eliminate the effects of self-moving objects[9]. In Hildreth's model, the visual field is divided into small regions and the region through which the most differential-vector lines pass is adopted as the first estimate. When differential vectors on each point are not oriented to the region, the point is considered as self-moving point. Finally her model estimates heading not using self-moving points. Although the method in this paper cannot estimate heading well when there are self-moving objects, the modifications of Hildreth are also applicable to this method.

We proposed another method for heading recovery [10]. Although roll is estimated in the same way, that method is different from the new algorithm presented in this paper. The algorithm in the earlier paper does not use the differential motion concept. The algorithm in our earlier paper recovers pitch and yaw first and then derived heading. On the other hand, the new algorithm recovers heading first and then derived the pitch and yaw parameters. It is worth noting that the algorithm in this paper is superior with respect to robustness to yaw and pitch compared with the method in the earlier paper although both methods assume small pitch and yaw. We performed simulations of both algorithms in the same conditions and found that the new algorithm is more robust to pitch and yaw. This advantage is preferable for tasks of computer vision.

Implication for biological visual system

Differential motion algorithms of Rieger & Lawton algorithm and the new method have a characteristic that they cannot recover heading when translation toward a frontoparallel plane is simulated which is also observed for human observers [10, 13, 14]. The methods might be used by the human brain to compute heading from motion information.

We presented another method for heading recovery in the earlier paper [10] as candidate for human model of heading judgment. In the earlier paper, we presented some results of perturbation stimuli, which are inconsistent with Rieger & Lawton algorithm. The results are also inconsistent with the method in this paper. However, we cannot say that the differential motion methods are not valid as biological model for only those results. The human visual system may use several different algorithms. Since each algorithm has some advantages relative to other algorithms, the visual system makes heading estimates more robust using several algorithms simultaneously. Further researches are required to determine which methods the human visual system actually uses.

It is not known that the human visual system uses the same methods as the visual system of other animals does. It is plausible that the visual system of monkey uses the same method for heading recovery because the human visual function is similar to monkey's. However, it is doubtful that all animals use the same methods as humans for heading recovery. There may exist some animals that use the algorithm in this paper. We think that the new algorithm is important for possible biological models of heading judgment.

Acknowledgement

M. Hanada was supported by Fellowships from the Japan Society for the Promotion of Science for Japanese Junior Scientists and Y. Ejima was supported by Special Coordination Funds for Promotion of Science and Technology Agency of the Japanese Government.

Author Contact Information:

Y. Ejima can be reached by e-mail at ejima@cv.jinkan.kyoto-u.ac.jp and M. Hanada hanada@cv.jinkan.kyoto-u.ac.jp

Reference

1. A. R Bruss, and B. K. P. Horn, “Passive navigation”, Comput. Graphics Image Process. **21**, 3-20 (1983).

2. H. C. Longuet-Higgins and K. Prazdny, “The interpretation of moving retinal images” Proc. R. Soc. London Ser. B **208**, 385-397 (1980).

3. K. Kanatani, “3-d interpretation of optical flow by renormalization”, Int. J. Comput. Vision **11**, 267-282 (1993).

4. C. Tomasi and J Shi, “Direction of heading from image deformations”, In Proceedings of IEEE Computer Vision and Pattern Recognition, 422-427 (1993).

5. J. H. Rieger and D. T. Lawton, “Processing differential image motion”, J. Opt. Soc.Am. A **2**, 354-360 (1985).

6. J.M.H. Beusmans “Computing the direction of heading from affine image flow”, Biol.Cybern. **70**, 123-136 (1993).

7. D. J. Heeger and A. Jepson, “Visual perception of three-dimensional motion” Neural Comp. **2**, 129-137 (1990).

8. D. J. Heeger and A. Jepson, “Subspace methods for recovering rigid motion 1: Algorithm and implementation”, Int. J. Comput. Vision **7**, 95-117 (1992).

9. E. C. Hildreth, “Recovering heading for visually-guided navigation”, Vision Res. **32**, 1177-1192 (1992).

10. M. Hanada and Y. Ejima “A model of human heading judgement in forward motion”, Vision. Res. (in press).

11. D. Regan and S. Kashal “Monocular Discrimination of the Direction of Motion in Depth”, Vision Res. **34**, 163-177 (1994).

12 Perrone, J. A. & Stone, L. S. "A model of self-motion estimation within primate extrastriate

visual cortex", Vision Res. 34, 2917-2938. (1994)

13. W. H. Warren, and D. Hannon, “Eye movement and optical flow”, J. Opt. Soc.Am. 7, 160-169 (1990).

14. C. S. Royden, J. A. Crowell and Banks, M. S. “Estimating heading during eye movements”, Vision Res. 34, 3197-3214 (1994).

Figure Captions

Figure 1: Center of outflow.

The center of outflow is the point which achieves the least square sum of d in (a). Examples of the center of outflow are shown in (b) and (c).

Figure 2: Examples of a flow filed and the virtual radial flow.

The flow field and the virtual radial flow are sketched in (a). Note that the flow is not actual one. The original flow and the virtual flow in Eq. (6) correspond to the projected motion of the original 3-D point and a frontoparallel plane with average depth of original sampling points' depth, as shown in (b). The difference vectors of the original and virtual velocity are oriented to the heading point (U/W, V/W) in the image.

Figure 3: Results in the ground condition.

Results of the simulation of the proposed algorithm in the ground condition are shown. The horizontal axis represents simulated heading (U/W) and the vertical axis represents heading estimated by the algorithm. Each point denotes the result of each trial.

Figure 4: Results in the cloud condition.

Results of the simulation of the proposed algorithm in the ground condition are shown. (a) The horizontal axis represents the horizontal component of simulated heading (U/W) and the vertical axis represents the value estimated by the algorithm. (b) The horizontal axis represents the horizontal component of simulated heading (V/W) and the vertical axis represents the value estimated by the algorithm. Each point denotes the result of each trial.

Figure 5: Effects of the number of dots.

Simulated heading and estimated one were divided into a horizontal component (U/W) and a vertical one (V/W). Slopes and correlation coefficients obtained by regression analyses conducted for each component are shown. The horizontal axis indicates the number of input dots.

Figure 6: Effects of the rotation rate.

Simulated heading and estimated one were divided into a horizontal component (U/W) and a vertical one (V/W). Slopes and correlation coefficients obtained by regression analyses conducted for each component are shown. The horizontal axis indicates the absolute value of pitch (A) and yaw (B) rates.

Figure 7: Analysis of split double-plane.

(a) Two planes with different depth are positioned in the upper and lower field, respectively. Focus 1 and 2 indicate the focus of the flow for each plane. The center of outflow is the best-fitting intersection of the line passing through all velocity vectors. The center of outflow is located between the two focuses. (b) The 3-D structure corresponding to the flow field in (a) is shown. The virtual radial flow corresponds to the image motion of the frontoparallel plane with average depth of the sampling points' depth.