1989

Optical Flow Estimation by Integrating Feature-Based with Flow-Based Schemes under Multiresolution

Chia-Hoang Lee
Zhongquan Wu

Report Number:
89-880

https://docs.lib.purdue.edu/cstech/748

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries. Please contact epubs@purdue.edu for additional information.
Optical Flow Estimation by Integrating Feature-Based with Flow-Based Schemes under Multiresolution

Chia-Hoang Lee
Zhongquan Wu

Department of Computer Sciences
Purdue University
West Lafayette, IN 47907

ABSTRACT

Despite great advances in the analysis of time-varying images, searching for correct and robust algorithm is still challenging and elusive. Two schemes, corresponding to small or large temporal intervals, are usually distinguished in measuring visual motion. One, called flow-based method, is based directly on the local intensity changes. The other one, called token-based method, is based on identifiable features which are located and matched over time. This paper presents a new scheme that combines a token-based technique with a multigrid flow-based technique to achieve a reliable estimation of optical flow field. Results with a sequence of real images are also provided.

Acknowledgement: This work was supported by the National Science Foundation under Grant IRI-8702053A1. Permanent address: Department of Information Electronics, Tsinghua University, China.
1. Introduction

Despite great advances in the analysis of time-varying images, searching for correct and robust algorithm is still challenging and elusive. Two schemes, corresponding to small or large temporal intervals, are usually distinguished in measuring visual motion. One, called flow-based method, is based directly on the local intensity changes. The other one, called token-based method, is based on identifiable features which are located and matched over time. For a comprehensive discussions, see [1].

One difficult and important problem in flow-based schemes for time-varying images is the estimation of optical flow. This intermediate representation of time-varying imagery assigns each pixel a flow vector describing its temporal displacement in the image plane or — for human vision — in the retinal field. Since only the normal component of the optical flow can be estimated, it thus is an ill-posed problem [2]. During the recent years, regularization have been proposed to overcome this difficulty. These techniques combine the fundamental flow equation with some other constraints e.g. the "smoothness constraint", "oriented smoothness constraint" etc. [3,4,5,6] to estimate the flow field. Although the regularization approach makes the problem well-posed, the computational aspect of estimating optical flow remains to be somehow unreliable. Many researchers have also used multiresolution strategy (coarse-to-fine) to improve such computational issues as the speed of convergence, and reliability of the solution[7].

As for the token-based schemes, there are two difficult problems: the correspondence problem and the structure from motion problem. The correspondence problem matches features in the successive images over time. If the temporal displacement of the features is large, then the combinatorial search might become unavoidable. Another difficulty of the token-based scheme is that most of the known techniques are quite sensitive to a small amount of noise in the images.

This paper describes an integration of the above two schemes for the time-varying imagery analysis. This new approach combines the bottom-up processing with the top-down processing to achieve a reliable flow field estimation. A feature-based technique is used for the bottom up processing, while the multigrid flow-based scheme is used for the top-down processing. In the bottom up step: features will first be matched, and a feature-based technique will be used to derive an approximate estimation of the motion parameters. Once an approximate estimation of the motion parameters is available, it will be used to predict the direction of the optical flow on a properly selected level of the image pyramid. Note that the magnitudes of the flow vectors can not be determined because the depths of nonfeatured points are unknown yet. On the other
hand, the normal flow – the normal component of the optical flow along the gradient direction of the intensity contour can be determined from the fundamental flow equation. Therefore, an approximate flow field can be derived by combining the normal flow with the predicted flow direction at pixels of a properly selected level from the two techniques. Next the top-down multigrid flow technique will use the profile-matching technique to refine the flow field. Consequently one could then refine the motion/structure estimation based on the derived dense velocity field. Figure 1 illustrates the framework of our approach.

2. An Integrated Scheme Under Multiresolution

Multigrid methods [8,9] have been applied to improve the estimation of optical flow. Here we also use multiresolution technique for refining the optical flow during the top-down processing. For the convenience of readers, it can be summarized as follows:

Step 1. Create the pyramids of an image sequence.

Step 2. Extract the features from the bottom levels of the pyramids. Match the extracted features and compute the motion parameters, i.e., $R$ – the instantaneous rotation, and $T$ – the instantaneous translation.

Step 3. Compute the optical flow based on $(R, T)$ and the normal flow estimated using the fundamental optical flow equation on the level properly selected. Refine the flow velocities by searching the best gray-tone profile matching around the direction of the velocities.

Step 4. Compute the flow velocities at every mesh point by using second order linear approximation model.

Step 5. Project the resulted flow velocities down to the next finer level as the initial velocities for searching the best gray-tone profile matching around the direction of the velocities of the top level.

Step 6. If the level $t \neq t_m$ ($t_m$ is the finest level ), Goto Step 4.

The flow chart of our approach is shown in Figure 3. Subsection 2.1 briefly presents the token-based schemes: what results have been obtained, and the associated computational issues. Subsection 2.2 presents the fundamental flow equation used in the flow-based technique. Subsection 2.3 estimates the flow vectors by combining the two schemes.
2.1. Feature-Based Schemes

The theoretical understanding of motion recovery from two images in feature-based schemes is well understood. For example, eight object points in the general positions will guarantee a unique recovery of motion and structure [10]. Another fact is that four coplanar points will yield at most two solutions [11,12]. However, most of the methods are quite sensitive to a small amount of noise in the image data.

In [13], we describe a computational procedure, based on a generate-and-test strategy, for visual motion computation. This approach consists of two steps. The first objective generates plausible solutions to account for a pair of sets of three image points irrespective of other features. The second objective tests these generated solutions to see if any one of them could account for features not considered in the first place. Consider an example of eight features in an image sequence as shown in figure 4(a)(b). This approach first forms 56 pairs of sets of three features. Based on each pair, our scheme derives a finite number of solutions (the number of solutions depends on the grid-size of the triangles defined by the three features). Next, these solutions are tested against the remaining five features to ensure its consistency. A solution accepted by these two steps is called admissible. The admissible solutions form then a distribution in a Hough parameter space where a cluster can be found containing the true solution. If the number of features is less than eight, then there may be more than one clusters. Our integrated approach uses this featured-based scheme for the bottom-up processing and assume that there exists always eight features in the images.

2.2. Flow-Based Schemes

[14,15,16] have shown that an exact solution for the system of partial differential equations can be given at the gray tone corners and extrema. These approaches assume a constant optical flow velocity within a small image region around the gray tone corner and extrema. [17] considers the edge displacement using the contour derived from zero-crossing. Since only one degree of freedom could be determined, a global criteria must also be introduced to bear for disambiguation. [18] propagate reliable velocity vectors on corners along the contours to resolve the ambiguity. However, Thompson and Kearney [6] analyzed the error source of the optical flow estimation and concluded that the local method is susceptible to a variety of problems. For example, the bad estimate often can not be distinguished from a good estimate, and the consequence is that the local method is often of little use. A remedy is to combine a coarse-to-fine analysis [19,20] with iterative registration to obtain a better result and increase the speed of convergence.
The fundamental optical flow equation [3] is used in this paper to compute the normal optical flow. I.e., the relationship

\[ \frac{\partial I}{\partial x} u_x + \frac{\partial I}{\partial y} u_y + \frac{\partial I}{\partial t} = 0 \]  

(2.2-1)

will be used to determine the normal component of the optical flow. \( \frac{\partial I}{\partial t} \) is the temporal intensity gradient, \( \frac{\partial I}{\partial x} \) and \( \frac{\partial I}{\partial y} \) are components of the spatial intensity gradient, and \( u_x, u_y \) are the \( x \) and \( y \) components of the flow velocity. From the above equation (2.2-1), we have the magnitude of the normal component

\[ V_n = -\frac{\partial I}{\partial t} \bigg/ |\nabla I| \]

where \( |\nabla I| \) is the magnitude of the spatial intensity gradient

\[ |\nabla I| = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2} \].

The normal flow field as well as the motion parameter \((R, T)\) will then be integrated by our approach described in the next section to compute the flow field.

2.3. Estimating the Flow Vectors by Combining the Two Schemes

Once \((R, T)\) and the normal flow on a coarser level of the image pyramid are available, our next goal is to estimate the flow vector.

Suppose a rigid body undergoes a motion in the half space. Let \( P(t) \) be the position vector of an object point at time \( t \). \( P(t) = (X(t), Y(t), Z(t))^T \) where the superscript \( T \) denotes transpose. Let \((x(t), y(t))\) denote the perspective projection of \( P(t) \) onto the image plane: \( Z = 1 \). The following equation describes the motion of an object point at two instants \( t_1 \) and \( t_2 \).

\[ P(t_2) = RP(t_1) + T \quad (2.3.1-1) \]

where

\[ P(t) = Z(t)(x(t), y(t), 1)^T = Z(t)p(t) \]

We may rewrite the Eq. (2.3.1-1) as follows

\[ Z(t_1) \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \end{bmatrix} \begin{bmatrix} x(t_1) \\ y(t_1) \\ 1 \end{bmatrix} + \begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{bmatrix} = Z(t_2) \begin{bmatrix} x(t_2) \\ y(t_2) \\ 1 \end{bmatrix} \quad (2.3.1-2) \]
where \( \gamma_1, \gamma_2 \) and \( \gamma_3 \) are the row vectors of the rotation matrix \( R \), and \( p(t) = (x(t), y(t), 1)^T \). Comparing the left and the right sides of (Eq.2.3.1-2) we have

\[
\begin{align*}
    x(t_2) &= \frac{Z(t_1)\gamma_1p(t_1) + \Delta X}{Z(t_1)\gamma_3p(t_1) + \Delta Z} \\
    y(t_2) &= \frac{Z(t_1)\gamma_2p(t_1) + \Delta Y}{Z(t_1)\gamma_3p(t_1) + \Delta Z}
\end{align*}
\]  

(2.3.1-3)

Because the estimated translation \( \vec{T} = (\vec{\tilde{t}}_1, \vec{\tilde{t}}_2, \vec{\tilde{t}}_3)^T \) is up to a scale factor of the real translation, it follows

\[
T = (\Delta X, \Delta Y, \Delta Z)^T = \alpha \vec{T} (\vec{\alpha} \vec{\tilde{t}}_1, \vec{\alpha} \vec{\tilde{t}}_2, \vec{\alpha} \vec{\tilde{t}}_3) \quad \text{for some scalar } \alpha.
\]

(2.3.1-4)

Substituting Eq. (2.3.1-4) to Eq. (2.3.1-3), we obtain

\[
\begin{align*}
    x(t_2) &= \frac{Z(t_1)\gamma_1p(t_1) + \vec{\alpha} \vec{\tilde{t}}_1}{Z(t_1)\gamma_3p(t_1) + \vec{\alpha} \vec{\tilde{t}}_3} \\
    y(t_2) &= \frac{Z(t_1)\gamma_2p(t_1) + \vec{\alpha} \vec{\tilde{t}}_2}{Z(t_1)\gamma_3p(t_1) + \vec{\alpha} \vec{\tilde{t}}_3}
\end{align*}
\]  

(2.3.1-5)

The basic optical flow equation is

\[
(\vec{u}_x(t_1) x(t_2) + \vec{u}_y(t_1) y(t_2) - (\vec{u}_x(t_1) x(t_1) + \vec{u}_y(t_1) y(t_1)) + V_n^2) = 0
\]

(2.3.1-6)

or

\[
Ax(t_2) + By(t_2) + C = 0
\]

for simplicity.

Combining the Eq. (2.3.1-5) and Eq. (2.3.1-6), we may express \( Z(t_1) \) as

\[
Z(t_1) = -\frac{\alpha(\vec{\alpha} \vec{\tilde{t}}_1 + B \vec{\tilde{t}}_2 + C \vec{\tilde{t}}_3)}{A \gamma_1p(t_1) + B \gamma_2p(t_1) + C \gamma_3p(t_1)} = -\frac{\alpha \phi(\vec{T})}{\phi(\vec{\gamma}p(t_1))}
\]

(2.3.1-7)

where \( \phi(\vec{T}) = \Delta \vec{\tilde{t}}_1 + B \vec{\tilde{t}}_2 + C \vec{\tilde{t}}_3 \) and \( \phi(\vec{\gamma}p(t_1)) = \Delta \gamma_1p(t_1) + B \gamma_2p(t_1) + C \gamma_3p(t_1) \).

Substituting Eq. (2.3.1-7) to Eq. (2.3.1-5), we obtain

\[
\begin{align*}
    x(t_2) &= \frac{\vec{\tilde{t}}_1 \phi(\vec{\gamma}p(t_1)) - \gamma_1p(t_1) \phi(\vec{T})}{\vec{\tilde{t}}_3 \phi(\vec{\gamma}p(t_1)) - \gamma_3p(t_1) \phi(\vec{T})} \\
    y(t_2) &= \frac{\vec{\tilde{t}}_2 \phi(\vec{\gamma}p(t_1)) - \gamma_2p(t_1) \phi(\vec{T})}{\vec{\tilde{t}}_3 \phi(\vec{\gamma}p(t_1)) - \gamma_3p(t_1) \phi(\vec{T})}
\end{align*}
\]  

(2.3.1-8)
and the displacements
\[
\begin{align*}
\Delta x &= x(t_2) - x(t_1) \\
\Delta y &= y(t_2) - y(t_1)
\end{align*}
\]

Suppose the time interval between any two frames is one unit, then
\[
\begin{align*}
\mu_x &= \Delta x/\Delta t = \Delta x \\
\mu_y &= \Delta y/\Delta t = \Delta y
\end{align*}
\]

2.4. Improving the Displacement Estimates Using Local Gray Value-Profile Matching

If magnitude $V_n$ of the normal flow (see figure 5) is not accurate enough, then it is often necessary to improve the magnitude of the velocity obtained. A gray-value-profile matching scheme (developed below) is used to refine the flow vector(displacement) field along the direction of the estimated vector at every grid point of the raw flow field. This idea relies on the robustness of the flow directions which in turn is determined by the accuracy of the featured-based scheme. If the flow directions are not good enough either, then an extended profile matching scheme will be necessary.

One approach to improve the estimate of the velocity magnitude of a point $t$ is to search the best match of $t$ in the second frame along its velocity direction. This can be done by comparing the gray-value-profile (g-v-p) of a local line window centered at $t$ with the gray-value-profile (measured in the second image) of a local line window centered at other points (see Figure 6). Let $g(s)$ be the g-v-p associated with $t$. Let $g_u(s)$ be the g-v-p associated with $u$. The following Chebyshev distance is taken for measuring the similarity between these two gray-value-profiles $g(s)$ and $g_u(s)$,

\[
d(u) = \max \{ g(s) - g_u(s) \}
\]

where the $[-s_0, s_0]$ represents the line window.

Before computing the Chebyshev distance of the two profiles, the third order polynomials are used to approximate the discrete gray values of $g(s)$ and $g_u(s)$. I.e. we may approximate $g(s)$ and $g_u(s)$ as follows.

\[
\begin{align*}
g(s) &= \alpha_3 s^3 + \alpha_2 s^2 + \alpha_1 s + \alpha_0 \\
g_u(s) &= \beta_3 s^3 + \beta_2 s^2 + \beta_1 s + \beta_0
\end{align*}
\]
This approximation reduces the high-frequency error substantially and the estimates of flow vectors can be improved by the subpixel matching. Since the above distance uses the uniformly convergence of the function, these two profiles are similar in a more global sense, so the error can be reduced substantially.

The distance between the two polynomials could be computed by locating the global minimum of \( \Delta g(s) = g(s) - g_u(s) \). The global minimum can only occur at one of the two end points \( -s_0, s_0 \) or one of the local extremum \( s_1, s_2 \). For the third order polynomial, it can be shown

\[
\begin{align*}
    s_1 &= \frac{b_2 - c_2 + \sqrt{(b_2 - c_2)^2 - 3(c_3 - b_3)(c_1 - b_1)}}{3(c_3 - b_3)} \\
    s_2 &= \frac{b_2 - c_2 - \sqrt{(b_2 - c_2)^2 - 3(c_3 - b_3)(c_1 - b_1)}}{3(c_3 - b_3)}
\end{align*}
\]

Therefore, \( d(u) \) could be computed and the best match can be found by locating \( u \) so that \( d(u) \) reaches its minimum.

If the feature-based scheme does not provide a good estimate of the motion parameters, then the profile-matchings have to be implemented not only along the directions of the raw vectors but also along some offset angles around the directions. One way is to search the best match of the two line windows within the interval \( [-\frac{\Delta \theta}{2}, \frac{\Delta \theta}{2}] \) around the direction. The flow field can then be improved by this extended profile-matching technique.

2.5. The Estimation of Velocities at Mesh Points

It is difficult to obtain a good normal flow at points where \( \partial I/\partial r \) is very large or \( |\nabla I| \) is very small. Our approach will not continue the computation at these points once they are detected in the early stage of computation. To interpolate the velocities at these points, a second order linear approximation model is used. The formula can be determined based on Taylor series expansion around the pixel. For details, see Appendix A.

3. The Experimental Results

As mentioned above, it is necessary to generate a sequence of image pyramids for implementing our scheme. In this section we will discuss the pyramid generation first, and then show the results obtained from experiments with real image sequence of the campus of an university.
3.1. The Pyramid Generation

Our method approximate a continuous problems using a pyramid of resolution levels \( t_0, t_1, \ldots, t_m \). The corresponding sizes are \( h_0 < h_1 < \cdots < h_m \). Assume that the grid points of each level are uniformly located and the ratio of the mesh sizes of adjacent levels is a multiple of two, i.e., \( h_k : h_{k+1} = 1:2 \).

A sequence of image pyramids can be generated by the following operator from a level to the next coarser level to reduce the resolution

\[
O_{t \to t-1} = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}
\]

The total number of levels for an image pyramid with a maximum of \( 256 \times 256 \) grid points is equal to \( \log_2(256) = 8 \). The level number for any middle level can be calculated by \( \log_2 \) (image size). The image pyramid in Figure 5 for the image in Figure 2.a. has been calculated with this operator. An appropriate level of the image pyramid has been selected as the start level \( t_s \) of our procedure. Here \( t_s = 5 \), because the displacement of the pixels on this level is about 2 pixels and the noise is substantially reduced after three \( O_{t \to t-1} \) operations.

Besides the data structure for the sequence of images \( g_1, g_2, \ldots, g_n \), the current vector field has to be transformed to a new grid level when the control of the multi-resolution is transferred to the next level of the pyramid. The inverse operator \( O_{t \to t+1} \) which transfers a vector field to a finer level implements a bi-cubic interpolation between the grid points of the coarser grid.

3.2. The Experimental Results

The experiments consists of real image sequence as shown in Figure 2. The 5-frame image sequence were taken of a model of campus building with camera mounted on a robot arm with the motion controlled by a computer. The image size is \( 256 \times 256 \) pixels.

Eight features and its matching over time are needed in our token-based scheme. While this step was discussed in [21], we have not yet automated it at this time. We identify the tokens and perform the matching manually. Figure 4 show the tokens and its matching. After running our scheme, wee obtain the following solution:
Table 3.1
The first row denotes translational vector.
The second row through the last row denotes rotational matrix.

<table>
<thead>
<tr>
<th></th>
<th>1.535367</th>
<th>1.866366</th>
<th>0.000000</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>0.998501</td>
<td>-0.002361</td>
<td>0.054677</td>
</tr>
<tr>
<td></td>
<td>0.004850</td>
<td>0.998955</td>
<td>-0.045450</td>
</tr>
<tr>
<td></td>
<td>-0.054512</td>
<td>0.045647</td>
<td>0.997469</td>
</tr>
</tbody>
</table>

Figure 5 shows one of the image pyramids in the image sequence. Figure 6 shows the image sequence corresponding to level 5 which has been chosen as the coarsest grid $l_s$ of the image pyramid.

The normal flow obtained by the fundamental optical flow equation is shown in Figure 7(a), and the flow field after applying our technique is shown in Figure 7(b). Although this flow field is not smooth enough, most of the flow vectors are close to the correct ones. A interpolation procedure was used to estimate the flow vectors at mesh points where the confidence of the estimated vectors are too low. Finally, the smoothness procedure terminated after 2 iterations on this coarsest level and the resultant flow field can be seen in Figure 7(c). The last flow field can be used for the flow estimation of the next finer level.

An example of the profile matching is shown in figure 12. The local gray-level structure of a line window centered at the pixel position $(10, 10)$ measured from the left upper corner in the frame 1 is shown in Figure 12(a). A sample of the local gray-level structures of the line windows centered at some positions along the flow direction in the frame 2 are displayed in Figures 12(b)-(g). All the local gray-level structures of these windows are fitted with the cubic polynomials as shown by the solid curves in Fig. 12 (a)-(g). From these Figures, it is clear that the curves in (d) and (e) with $\delta = 2$ and $\delta = 3$ are most similar to the curve in (a). A subpixel matching is then performed to find the best match at $\delta = 2.3$. This means that the magnitude of the velocity is 2.3 and the direction is $45^\circ$.

The initial flow field for the next finer grid level 6 has been calculated from the flow field in Figure 7(c) using the interpolation operator $O_{t \rightarrow t+1}$ and the resultant flow
field is shown in Figure 9(a). The image sequence for grid level 6 is shown in Figure 8. After the profile-matching, a little improvement can be seen in the flow field of Figure 9(b) because the initial estimation of the flow field is close to the correct one. Both of the two flow field are smooth enough, so the smooth procedure terminates.

The flow field for the image sequence of grid level 7 (see Figure 10) is obtained from Figure 11 by using the bicubic interpolation. Figure 12 shows the final flow field for the image sequence at the finest grid level. Only the bicubic interpolation is necessary to terminate the coarse-to-fine procedure on the finest grid level beginning with the coarser flow shown in Figure 11.

4. Conclusion

In this paper, we have presented an integration scheme that combines the feature-based technique with the multigrid flow-based scheme. The feature-based technique derives an approximate motion parameters and then estimates the image flow direction. The flow-based technique provides the normal component of the flow field. By integrating these two information, one obtains a dense flow field. The advantage contains: (1) It is easier to achieve a reliable estimation of the optical flow vectors, (2) The convergence speed can be improved significantly, (3) Even for the displacements that are relatively large compared to the size of the operator masks, acceptable optical flow vectors can be computed by the profile-matching algorithm with reasonable accuracy.
Reference


13. C. H. Lee, "Towards Implementing Robust Visual Motion Computations: Generate and Test Approach" Technical Reports, Department of Computer Sciences, Purdue


17. E. C. Hildreth, "Computing the Velocity along Contours", Proceed. of Workshop on Motion: Representation and Perception, Toronto, Canada, April 4-6, 26-32, 1983.


a sequence of image pyramid

Figure 1. A block diagram of the combinational scheme.
Figure 2.
A five-frame image sequences. The order of the sequence (a)-(d) is from top to bottom and left to right.
Figure 3. The flow chart of our scheme.
Figure 4.
The two image frames with the marked features.
Figure 5. The normal component and the flow velocity.
Figure 6.
(a) The g-v-p (gray-value-profile) of the window centered at (10,10) in frame 1. (b)-(f) The g-v-p of the window with the displacement of the $\delta = 0, 1, 2, 3, 4$, respectively, of the center from (10,10) in frame 2. (g) The solid-line is the same as the g-v-p in (a) and the dash-line is the same as the g-v-p in (d) but with $\delta = 2.3$. 
Figure 7.
Image pyramid corresponding figure 2(e) in the image sequence. The sizes are 256x256, 128x128, 64x64, 32x32 respectively. For visual purpose, we have magnify the size of each level to 256x256.
Figure 8.
The image sequence of the level 5 of the pyramids corresponding to Figure 2(c)-(e).
Figure 9.
Flow fields for level 5. (a) A normal flow has been used as the initial vector flow. (b) The flow field after the combinatorial process and profile matching. (c) The resultant flow field after 2 iterations of smoothing.
Figure 10.
The image sequence for the level 6 of the pyramids corresponding to Figure 2(c)-(e).
Figure 11. Flow fields for the level 6. (a) The flow field obtained using bicubic interpolation from the previous coarser level 5. (b) The resultant flow field after the profile-matching.
Figure 12.
The image sequence for the level 7 of the pyramids corresponding to Figure 2(c)-(e).
Figure 13. Flow fields for the level 7 of the image sequence in Figure 12.
Appendix A. The Interpolation of Velocities at Mesh Points

The following second order linear approximation model is applied to compute the velocities at mesh points.

Suppose \( P_o(x_o, y_o) \) is the coordinates of a mesh point, and \( P(y_k, y_k) \) is a neighboring point of \( P_o \) with the known velocity. Let

\[
\Delta x_k = x_k - x_o \\
\Delta y_k = y_k - y_o
\]

Then based on Taylor series expansion we have:

\[
u(x_k, y_k) = u(x_o, y_o) + \frac{\partial u}{\partial x} \bigg|_{P_o} \Delta x_k + \frac{\partial u}{\partial y} \bigg|_{P_o} \Delta y_k + \frac{\partial^2 u}{\partial x \partial y} \bigg|_{P_o} \Delta x_k \Delta y_k + \frac{\partial^2 u}{\partial x^2} \bigg|_{P_o} \Delta x_k^2 + \frac{\partial^2 u}{\partial y^2} \bigg|_{P_o} \Delta y_k^2
\]

\[
v(x_k, y_k) = v(x_o, y_o) + \frac{\partial v}{\partial x} \bigg|_{P_o} \Delta x_k + \frac{\partial v}{\partial y} \bigg|_{P_o} \Delta y_k + \frac{\partial^2 v}{\partial x \partial y} \bigg|_{P_o} \Delta x_k \Delta y_k + \frac{\partial^2 v}{\partial x^2} \bigg|_{P_o} \Delta x_k^2 + \frac{\partial^2 v}{\partial y^2} \bigg|_{P_o} \Delta y_k^2
\]

Where \( u(x_k, y_k), v(x_k, y_k) \) are the \( x \) and \( y \) components of the nonzero velocity at some mesh points. They have been obtained in the previous step.

Solving equations (A-2) and (A-3) we may estimate the velocity and its partial derivatives at each mesh point \( P_o \) by means of the least-squared method.