A MULTISCALE TECHNIQUE FOR OPTICAL FLOW COMPUTATION USING PIECEWISE AFFINE APPROXIMATION

Ha V. Le, Guna Seetharaman

Center for Advanced Computer Studies
University of Louisiana at Lafayette
Lafayette, LA 70504-4330

Bertrand Zavidovique

Institut d’Electronique Fondamentale, Bat. 220
Université Paris XI, 91405 Cedex, France

Abstract

We present a technique to estimate the optical flow in an image sequence, based on a piecewise affine model. In this piecewise approach, the area of interest in each image frame is divided into a set of small triangular patches. These triangular meshes are established over a set of feature points, which are extracted from the images and tracked from one frame to another. The velocity field within each triangular patch is parameterized by an affine transform. A multiscale coarse-to-fine approach is employed to increase the robustness of the method as well as the accuracy of the optical flow resulted from piecewise affine approximations. Finally, an adaptive filter is used to refine the estimated flow field. The filter is designed in such a way that not only can it reduce noises caused by errors of the process described above, but it can also avoid smoothing the discontinuities in the motion field. The method has been implemented and some experimental results are presented in this paper. The method takes advantage of widely used MPEG-4 encoding hardware/software tools.

1. INTRODUCTION

An optical flow is a dense 2-D vector field, which features the motions in a dynamic scene projected onto an image plane. Optical flow-based methods make up one of the two main paradigms for motion estimation. The other paradigm, called the feature-based correspondence approach, seeks to recover motion parameters by tracking the motion of a set of image features (points, lines, etc.) along the image sequence. Thank to their ability to describe the motion at every pixel in the image, the optical flow-based approach is more robust than the feature-based correspondence methods when dealing with complex cases such as non-rigid motions.

Most optical flow estimation methods belong to a category called the differential approach. Differential methods use local spatio-temporal derivatives to estimate the instantaneous velocity vector at every pixel in the image. There is a class of the differential approach, called the feature-based techniques, in which the flow is first computed at the features in the image, then propagated to other pixels. Edges and corners are the most commonly-used features. It has been known that the optical flow estimated by differential techniques are most accurate at boundaries and corners [1, 2]. One of the problems for the differential methods is that some components of the motion field such as those along the isobrightness contours can not be observed at local scale. This is the so-called aperture problem. A common approach to overcome this problem is to apply some smoothness constraint to regularize the flow field [3, 4, 5, 6].

Correlation-based methods make up another class of optical flow techniques. This is a matching approach, which recovers the velocity vector field by finding correspondences of pixels between adjacent image frames. Determining the velocity vectors by matching is computationally expensive if performed at all pixels in the image. Besides, the correlation-based methods are only well-conditioned at and near feature points like corners. Therefore, they are usually used on as a part of other methods [6].

Parametric models have been used by many authors to describe the optical flows. Parametric model-based methods need to make explicit assumptions about the motion and objects in the scene. The optical flow can be modeled as a parametric function of pixel coordinates. This is a convenient way to describe the optical flow, but finding a precise model for the motion is a difficult task if there is no a priori knowledge about the motion and the scene. The commonly-used models are constant-flow models [7], affine models [8, 9, 10, 11, 12], and quadratic models [5, 13]. There are global models and piecewise models. While the use of global models is generally limited to rigid motions, piecewise models are often considered when dealing with non-rigid motions because of their robustness [10, 11, 12]. Black and Anandan [14] proposed a robust estimation framework which combines the recovery of multiple parametric motions and regularization techniques to recover piecewise-smooth flow fields. Experiments showed their robust estimation method can produce better results than most other
optical flow computation techniques. However, the computational cost can be very high.

Our objective is to deliver a motion estimation technique which is reliable, yet simple enough to be implemented in hardware/software for video processing using such standards as MPEG-4. Our method follows a framework proposed by Cho et al. [11] for optical flow computation based on a piecewise affine model. A surface moving in the 3-D space can be modeled as a set of small planar surface patches so that projected motion of each of those 3-D planar patches in a 2-D plane between two consecutive image frames can be described by an affine transform. Basically, this is a mesh-based technique for motion estimation, using 2-D content-based meshes. The advantage of content-based meshes over regular meshes is their ability to reflect the content of the scene by closely matching boundaries of the patches with boundaries of the scene features [10], yet finding feature points and correspondences between features in different frames is a difficult task. A multiscale coarse-to-fine approach is utilized in order to increase the robustness of the method as well as the accuracy of the affine approximations. An adaptive filter is used to smooth the flow field such that the flow appears continuous across the boundary between adjacent patches, while the discontinuities at the motion boundaries can still be preserved. Many of these techniques are already available in MPEG-4.

2. THE METHOD

Our optical flow computation method includes the following phases:

1. Feature extraction and matching: in this phase the feature points are extracted and feature matching is performed to find the correspondences between feature points in two consecutive image frames.

2. Piecewise flow approximation: a mesh of triangular patches is created, whose vertices are the matched feature points. For each triangular patch in the first frame there is a corresponding one in the second frame. The affine motion parameters between these two patches can be determined by solving a set of linear equations formed over the known correspondences of their vertices. Each set of these affine parameters define a smooth flow within a local patch.

3. Smoothing: this is the refining phase, in which the flow field is smoothed with an adaptive filter.

2.1. The Multiscale Approach

Affine motion is a feature of the parallel projection, yet it is common even in applications using the perspective imaging model to use a 2-D affine transform to approximate the 2-D velocity vector field produced by a small planar surface patch moving rigidly in the 3-D space, since the quadratic terms of the motion in such a case are very small. A curved surface can be approximated with a set of small planar surface patches, then the motion of the curved surface can be described by a piecewise set of affine transforms, one for each planar patch, even if the surface is non-rigid, because a non-rigid surface can be approximated with a set of small rigid patches. The more number of patches are used, the more accurate the approximation is. Therefore, it is obvious that we would want to create the mesh in each image frame using as many feature points as possible. The problem is, when the set of feature points in each frame is too dense, finding correspondences between points in two consecutive frames would be very difficult, especially when the displacements are relatively large.

Our solution for this problem is a multiscale scheme. It starts at a coarse level with only a few feature points, so matching them would be fairly simple. A piecewise set of affine motion parameters, which gives an approximation of the motion field, is computed from these matching points. At the next finer scale, more feature points are extracted. Each of the feature points in the first frame has a target in the second frame, which is given by an affine transform estimated in the previous iteration. To find a potential match for a feature point in the first frame, the algorithm has to consider only those feature points in the second frame, which are close to its target point. This iterative process guarantees convergence, i.e. the errors of the piecewise affine approximations get smaller after each iteration.

2.2. Feature Point Extraction

As we mentioned earlier, edge and corner points are the most commonly-used features for motion estimation methods which require feature matching. It is due to the availability of numerous advanced techniques for edge and corner detection. Besides, it has been known that most of the optical flow methods are best-conditioned at edges and edge corners. We follow the suit by looking for points located at curved parts (corners) of edges. Edge points are identified first by using Canny edge detection method. Canny edge detector [15] applies a lowpass filter on the input image, then performs non-maxima suppression along the gradient direction at each potential edge point to produce thin edges. Note that the scale of this operation is specified by the width \( \sigma_e \) of the 2-D Gaussian function used to create the lowpass filter. Using a Gaussian with a smaller value of \( \sigma_e \) means a finer scale, giving more edge points and less smooth edges. To find the points located at highly-curved parts of the edges, a curvature function introduced by Mokhtarian and Mackworth [16] is considered. Their method allows the curvature measurement along a 2-D curve \( \Gamma(s) = (x(s), y(s)) \), \( s \) is
the arc length parameter, at different scales by first convolving the curve \( \Gamma \) with an 1-D Gaussian function \( g(s, \sigma_k) = \frac{1}{\sigma_k \sqrt{2\pi}} e^{-\frac{s^2}{2\sigma_k^2}} \), where \( \sigma_k \) is the width of the Gaussian.

\[
\mathcal{X}(s, \sigma_k) = \int_{-\infty}^{+\infty} x(s_1)g(s-s_1, \sigma_k)ds_1
\]

\[
\mathcal{Y}(s, \sigma_k) = \int_{-\infty}^{+\infty} y(s_1)g(s-s_1, \sigma_k)ds_1
\]

The curvature function \( \kappa(s, \sigma_k) \) is given by

\[
\kappa(s, \sigma_k) = \frac{\mathcal{X}(s, \sigma_k)\mathcal{Y}''(s, \sigma_k) - \mathcal{X}''(s, \sigma_k)\mathcal{Y}(s, \sigma_k)}{[\mathcal{X}(s, \sigma_k)^2 + \mathcal{Y}''(s, \sigma_k)^2]^{3/2}}
\]  

(1)

The first and second derivatives of \( \mathcal{X}(s, \sigma_k) \) and \( \mathcal{Y}(s, \sigma_k) \) can be obtained by convolving \( x(s) \) and \( y(s) \) with the first and second derivatives of the Gaussian function \( g(s, \sigma_k) \), respectively. The feature points to be chosen are the local maxima of \( |\kappa(s, \sigma_k)| \) whose values must also exceed a threshold value \( t_k \). At a finer scale, a smaller value of \( \sigma_k \) would be used, resulting in more corner points to be extracted.

2.3. Feature Point Matching

Finding the correspondences between feature points in consecutive frames is the key step of our method. We devised a matching technique in which the cross-correlation, curvature, and displacement are used as matching criteria. The first step is to find an initial estimate for the motion at every feature point in the first frame. Some matching techniques such as that in [17] have to considered all possible pairs, hence \( M \times N \) pairs needed to be examined, where \( M \) and \( N \) are the number of feature points in the first and second frames, respectively. Some others assume the displacements are small to limit the search for a match to a small neighborhood of each point. By giving an initial estimate for the motion at each point, we are able to reduce the number of pairs to be examined without having to constrain the motion to small displacements. Remember that we are employing a multiscale scheme, in which the initial estimation of the flow field at one scale is given by the piecewise affine transforms computed at the previous level, as mentioned in 2.1. At the starting scale, a rough estimation can be made by treating the points as if they are under a rigid 2-D motion. It means the motion is a combination of a rotation and a translation. Compute the centers of gravity, \( C_1 \) and \( C_2 \), the angles of the principal axes, \( \alpha_1 \) and \( \alpha_2 \), of the two sets of feature points in two frames. The motion at every feature points in the first frame can be roughly estimated by a rotation around \( C_1 \) with the angle \( \phi = \alpha_2 - \alpha_1 \), followed by a translation represented by the vector \( t = x_{C_2} - x_{C_1} \), where \( x_{C_1} \) and \( x_{C_2} \) are the vectors representing the coordinates of \( C_1 \) and \( C_2 \) in their image frame.

Let \( i^t \) and \( j^{t+1} \) be two feature points in two frames \( t \) and \( t+1 \), respectively. Let \( d(i^t, j^t) \) be the estimated match of \( i^t \) in frame \( t+1 \), \( d(i^t, j^{t+1}) \) be the Euclidean distance between \( i^t \) and \( j^{t+1} \), \( c(i, j) \) be the cross-correlation between \( i^t \) and \( j^{t+1} \), \( 0 \leq c(i, j) \leq 1 \), and \( \Delta \kappa(i, j) \) be the difference between the curvature measures at \( i^t \) and \( j^{t+1} \). A matching score between \( i^t \) and \( j^{t+1} \) is defined as follows

\[
d(i^t, j^t) > d_{\text{max}} : 
\begin{align*}
\text{score}(i, j) &= w_c c(i, j) + s_k(i, j) + s_d(i, j),
\end{align*}  
\]  

(2)

where

\[
\begin{align*}
s_k(i, j) &= w_k (1 + \Delta \kappa(i, j))^{-1} \\
s_d(i, j) &= w_d (1 + d(i^t, j^{t+1}))^{-1}
\end{align*}
\]  

(3)

The quantity \( d_{\text{max}} \) specifies the maximal search distance from the estimated match point. \( w_c \), \( w_k \), and \( w_d \) are the weight values, determining the importance of each of the matching criteria. The degree of importance of each of these criteria changes at different scales. At a finer scale, the edges produced by Canny edge detector become less smooth, meaning the curvature measures are less reliable. Thus, \( w_k \) would be reduced. On the other hand, \( w_d \) would be increased, reflecting the assumption that the estimated match becomes closer to the true match. For each point \( i^t \), its optimal match is a point \( j^{t+1} \) such that \( \text{score}(i, j) \) is maximal and exceeds a threshold value \( t_s \).

2.4. Affine Flow Computation

Consider a planar surface patch moving under rigid motion in the 3-D space. In 2-D affine models, the change of its position in frames \( t \) and \( t+1 \) is approximated by an affine transform

\[
\begin{bmatrix}
x^{t+1} \\
y^{t+1}
\end{bmatrix} = \begin{bmatrix}
a & b \\
c & d
\end{bmatrix} \begin{bmatrix}
x^t \\
y^t
\end{bmatrix} + \begin{bmatrix}
e \\
f
\end{bmatrix},
\]  

(4)

where \( (x^t, y^t) \) and \( (x^{t+1}, y^{t+1}) \) represent the coordinates of a moving point in frames \( t \) and \( t+1 \), \( a, b, c, d, e, \) and \( f \) are the affine transform parameters. Let \( \mathbf{x} \) be vector \( [x, y]^T \). The point represented by \( \mathbf{x} \) is said to be under an affine motion from \( t \) to \( t+1 \). Then the velocity vector \( \mathbf{v} = [dx/dt, dx/dt]^T \) of that point at time \( t \) is given by

\[
\mathbf{v}^t = \mathbf{x}^{t+1} - \mathbf{x}^t = \begin{bmatrix}
a - 1 & b \\
c & d - 1
\end{bmatrix} \mathbf{x}^t + \begin{bmatrix}
e \\
f
\end{bmatrix}
\]  

(5)

A and \( \mathbf{c} \) are called the affine flow parameters.

Using the constrained Delaunay triangulation [18] for each set of feature points, a mesh of triangular patches is
generated to cover the moving part in each image frame. A set of line segments, each of which connects two adjacent feature points on a same edge, is used to constrain the triangulation, so that the generated mesh would closely matches the true content of the image. From (5), two linear equations of six unknowns are formed for each pair of corresponding feature points. Therefore, for each pair of matching triangular patches, a total of six linear equations is established from their corresponding vertices. Solving these equations we obtain the affine motion parameters, which define the affine flow within the small triangular region.

2.5. Smoothing the Flow

Significant errors in the optical flow resulted from the piecewise affine approximations can appear due to such reasons as: 1) Mismatches between feature points, or 2) Patches stretching across several parts of the motion field whose actual velocity vectors are disconnected. The second case often occurs when dealing with non-rigid motions. One solution is to apply some regularization constraints on the flow field to regularize the noise. The regularization methods are more likely to converge, but are computationally expensive since they require solving optimization problems. Another option is to employ noise filtering techniques. Spatial filters can be used to smooth the flow, but they must take into consideration the discontinuities in the motion field, which need to be preserved. Temporal filtering techniques such as Kalman filtering can also be used. Techniques utilizing the filtering approach are fast in general, and therefore can be applied in systems that require real-time processing.

Our adaptive filter is a spatial filtering technique, which is based on the following assumptions

1. The intensity of a moving point changes little between two consecutive frame.
2. Discontinuities in the motion field occur only at the discontinuities in the image brightness, i.e. the velocity vector field over neighboring pixels which have similar intensities should be continuous.

Given these assumptions, the flow field can be smoothed as follows

\[
\mathbf{v}_{i}^{t} = \sum_{j \in N_i} g(\Delta I_t(j), \sigma_t) g(\Delta I_t(i, j), \sigma_s) \mathbf{v}_{j}^{t},
\]

where \(\mathbf{v}_{i}^{t}\) and \(\mathbf{v}_{j}^{t}\) are the velocity vectors of points \(i\) and \(j\) at time \(t\), \(N_i\) is a neighborhood of point \(i\), \(g(x, \sigma)\) is the 1-D Gaussian function with the width \(\sigma\), \(\Delta I_t(j)\) is the temporal change in intensity of a moving point \(j\) between two frames \(t\) and \(t+1\), and \(\Delta I_t(i, j)\) is the difference between the intensities of points \(i\) and \(j\) in frame \(t\).

\[
\begin{align}
\Delta I_t(j) &= |I_t^{t+1} - I_t^j| \\
\Delta I_t(i, j) &= |I_t^i - I_t^j|,
\end{align}
\]

in which \(I_t^i\) is the intensity of point \(i\) in frame \(t\), and so on.

3. EXPERIMENTAL RESULTS

We conducted experiments with our technique using some popular test sequences for optical flow techniques and compared the results with those in [19] and [14]. The image sequences we used for the purpose of error evaluation include the Translating Tree sequence, the Diverging Tree sequence, and the Yosemite sequence. We also tested the algorithm on some real image sequences such as the NASA’s Coca-Cola sequence, the SRI Tree sequence, and the Rubic sequence.

For the Translating Tree and Diverging Tree sequences, the performance of our technique is better or comparable to most other methods shown in [19]. The lack of features led to large errors at some parts of the images in these two sequences, especially in the Diverging Tree sequence, increasing the average errors significantly, even though the estimated flow fields are accurate for most parts of the images.

The Yosemite sequence is a complex test. There are diverging motion due to the movement of the camera and translating motions of the clouds. While all the techniques analyzed in [19] showed significant increases of errors in
comparison with the results from the previous two sequences, the performance of our technique remained consistent.

4. CONCLUSIONS

A multiscale method for optical flow estimation, based on a piecewise affine model, has been presented. The technique is fairly simple with the use of the linear affine approximation, yet it is robust and able to produce accurate approximations, thanks to the multiscale piecewise approach. Feature matching plays an essential role in the process, with curvature measures of edge corners being used as a matching criterion, in addition to cross-correlation and initial estimates of the motion. We also introduced an adaptive filtering technique, which can be considered as a simple method to smooth the flow field without having to use regularization constraints. The filter was designed in such a way that it can reduce errors, but still preserve the discontinuities in the motion field. This optical flow estimation technique has been implemented and our experiments showed that it has worked as expected.

5. REFERENCES


Fig. 4. Two frames of the Translating Tree sequence

Fig. 5. Triangular meshes

Fig. 6. The correct flow (left) and the estimated flow (right)
Fig. 7. Two frames of the Diverging Tree sequence

Fig. 8. Triangular meshes

Fig. 9. The correct flow (left) and the estimated flow (right)
Fig. 10. Two frames of the Yosemite sequence

Fig. 11. Triangular meshes

Fig. 12. The correct flow (left) and the estimated flow (right)
Fig. 13. An estimated flow field from the NASA’s Coca-Cola sequence

Fig. 14. An estimated flow field from the SRI’s Tree sequence

Fig. 15. An estimated flow field from the Rubic sequence