

Vector Field Resampling using Local Streamline Approximation

M. Thomas, C. Kambhamettu
*Video/Image Modeling and Synthesis Lab,
Dept. of Computer and Info. Sciences,
University of Delaware,
{mani, chandrak}@udel.edu*

C. A. Geiger
*Center for Climatic Research,
Dept. of Geography,
University of Delaware,
cgeiger@udel.edu*

Abstract

In this paper, we propose an algorithm to resample coarse vector fields in order to obtain vector fields of a higher density. Unlike the typical linear interpolation scheme, our algorithm attempts to identify streamline characteristics in the flow field, and uses local polynomial parameterization of the flow to perform interpolation. Quantitative validation of the streamline oriented algorithm indicate that the Mean Square Error of the flow field obtained is more accurate than those obtained by bilinear interpolation schemes.

1. Introduction

Estimation of the apparent motion between time-lagged pairs of images has been an active area of research in computer vision. The algorithms that have evolved over the years can be roughly classified into 1) feature based, 2) gradient based, and 3) region based - depending on the way the motion is estimated. Feature based techniques can estimate large motion but provide sparse measurements. Gradient based mechanisms compute dense motion but are accurate only under the assumption of extremely small motion. Region based techniques provide an intermediate level of motion estimation density as the computations typically occur at specific grid locations on the image ([1] and references therein). In this last case, the estimation proceeds by assuming that the motion of pixels between grid locations are uniform. Due to the ease of hardware implementation and the parallelizability of region based techniques, these techniques are predominant among present day video encoders [4].

The biggest drawback of the region based scheme is that motion is assumed to be uniform between grid locations, which might not hold up in reality. The simplest means of obtaining motion at non-grid locations is by

linearly resampling the motion from the grid locations [5]. By repeating this process across all grid locations we can obtain a field that is denser than the original flow field. For color pixels, bilinear interpolation is typically used as a reasonably accurate estimate at non-grid locations. However, when resampling motion, the flow specific characteristics typically tend to show a higher correlation along the direction of flow lines than in any other direction. Taking these characteristics into consideration when interpolating a flow field might help improve the estimation accuracy. In this paper we attempt to describe an approach to interpolate flow fields by capturing flow specific characteristics. We also show that our algorithm produces a lower Mean Square Error (MSE) over bilinear interpolation, especially in the presence of noise.

The organization of the paper is as follows. We first describe the streamline algorithm that we developed. Subsequently we apply our algorithm to synthetic data sets to measure the accuracy of estimation under varying noise parameters and finally conclude with possible future directions.

2. Description of the Algorithm

The basic premise for the algorithm to estimate particle streamlines comes from the Line Integral Convolution (LIC) algorithm proposed by Lee and Cabral [2]. LIC was initially proposed as a tool to visualize high resolution vector fields by computing the streamlines from the vector fields. In the visualization scheme, the streamlines were convolved with white noise such that the correlation was only maintained along the flow.

The main goal of our algorithm is to compute local polynomial approximations to streamlines using each grid location as an anchor point. This allows the streamline contours to be extracted within a curvilinear coordinate system along the flow direction. For the purpose of our computation, we used the distance from the anchor

point as the parameter to specify the curvilinear coordinate system. Using this parameterization, we defined the polynomial approximation and its derivative as

$$x(s) = \sum_{k=0}^N a_k s^k \quad x'(s) = \sum_{k=1}^N k a_k s^{k-1} \quad (1)$$

$$y(s) = \sum_{k=0}^N b_k s^k \quad y'(s) = \sum_{k=1}^N k b_k s^{k-1} \quad (2)$$

where N is the order of the polynomial and s is the ‘‘anchor length parameterization’’. The main advantage of using this parameterization is that the direction of the streamline at the anchor location ($s = 0$) can be easily computed as

$$\theta = \tan^{-1} \left[\frac{y'(0)}{x'(0)} \right] = \tan^{-1} \left[\frac{b_1}{a_1} \right] \quad (3)$$

and the corresponding unit tangent would be $\mathbf{e} = [\cos(\theta), \sin(\theta)]$. To obtain the motion field at a higher resolution, we resample the polynomial coefficients, b_1 and a_1 to compute the directional components at non-grid locations.

An overview of our streamline computation algorithm is provided in Algorithm 1. The readers are urged to note that $(x, y) \leftarrow (x, y) \pm \mathbf{e}_{\Omega}^{[x],[y]}$ updates the current location by taking a unit integer (‘‘round operation’’) step in the positive or negative direction.

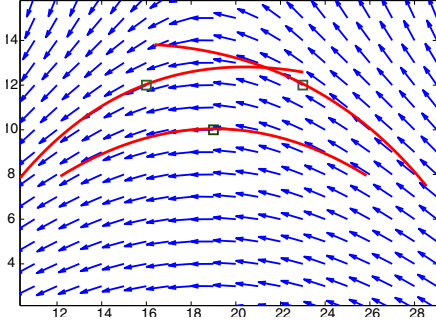


Figure 1. Computation of streamlines using sample anchor points.

Figure 1 shows the process of estimating the streamline contour for a sample set of anchor points (small dark squares). In our experiments, we used a 2^{nd} and 3^{rd} order polynomial for trajectory parameterization, which appeared reasonable enough for our purposes. The red lines in the figure are 2^{nd} order polynomials that define the most plausible flow direction at the corresponding black squares.

Algorithm 1 Computation of Vector Fields Streamlines

Input: $\mathbf{u}_{\Omega} \in \mathbf{R}^{h \times w \times 2}$
 $\mathbf{e}_{\Omega}^{ij} \leftarrow \mathbf{u}_{\Omega}^{ij} / \|\mathbf{u}_{\Omega}^{ij}\|$, $(i, j) = \{1 \cdots h\} \times \{1 \cdots w\}$
 $\mathcal{L} \leftarrow$ Maximum Length of the Integral Curve
for all (r, c) such that $r = 1 \cdots h, c = 1 \cdots w$ **do**
 $x \leftarrow c, y \leftarrow r, \mathcal{B} \leftarrow [x, y, 0]$
 {Forward Direction +, Backward Direction -}
for $k = 1$ to $\mathcal{L}/2$ **do**
 $(x, y) \leftarrow (x, y) \pm \mathbf{e}_{\Omega}^{[x],[y]}$
if (x, y) is outside Ω **then**
 break
end if
 append $[x, y, \pm \sqrt{(x-c)^2 + (y-r)^2}]$ to \mathcal{B}
end for
 Sort the rows of \mathcal{B} using column 3
 $s \leftarrow \mathcal{B}[:, 3], x(s) \leftarrow \mathcal{B}[:, 1], y(s) \leftarrow \mathcal{B}[:, 2]$
 $x(s) = \sum_{k=0}^N a_k s^k$ and $y(s) = \sum_{k=0}^N b_k s^k$
for $k = 0$ to N **do**
 $A(r, c, k) \leftarrow a_k, B(r, c, k) \leftarrow b_k$
end for
end for
 Interpolate A, B and $\|\mathbf{u}_{\Omega}\|$
for all (r, c) such that $r = 1 \cdots h, c = 1 \cdots w$ **do**
 $\theta(r, c) \leftarrow \arctan \left[\frac{B(r, c, 1)}{A(r, c, 1)} \right]$
end for
Output: $\|\mathbf{u}_{\Omega}\| \otimes \cos \theta$ and $\|\mathbf{u}_{\Omega}\| \otimes \sin \theta$ where \otimes is element-wise multiplication.

3. Results and Analysis

We computed the accuracy of our algorithm with two types of synthetic flow fields. The first set of parametric flow fields were computed as a function of the grid locations while the second set were obtained from a solution to the Navier Stokes equation of a ‘‘jet stream’’ flow. The second set of images form the standard flow fields that are used in measuring motion estimation accuracy for Particle Image Velocimetry imagery [6].

3.1. Synthetic Flow Fields

The three synthetic flow fields that we used to compare our algorithm against are

$$u_{ij} = y_{ij} \quad v_{ij} = x_{ij} \quad (4)$$

$$u_{ij} = x_{ij} \cos\left(\frac{y_{ij}}{\mathcal{K}}\right) \quad v_{ij} = y_{ij} \sin\left(\frac{y_{ij}}{\mathcal{K}}\right) \quad (5)$$

$$\begin{bmatrix} u_{ij} \\ v_{ij} \end{bmatrix} = \begin{bmatrix} -0.45 & -0.55 \\ 0.85 & 0.15 \end{bmatrix} \begin{bmatrix} x_{ij} \\ y_{ij} \end{bmatrix} \quad (6)$$

where x_{ij} and y_{ij} were the grid coordinates for the flow field. In our experiments we used $x_{ij}, y_{ij} \in$

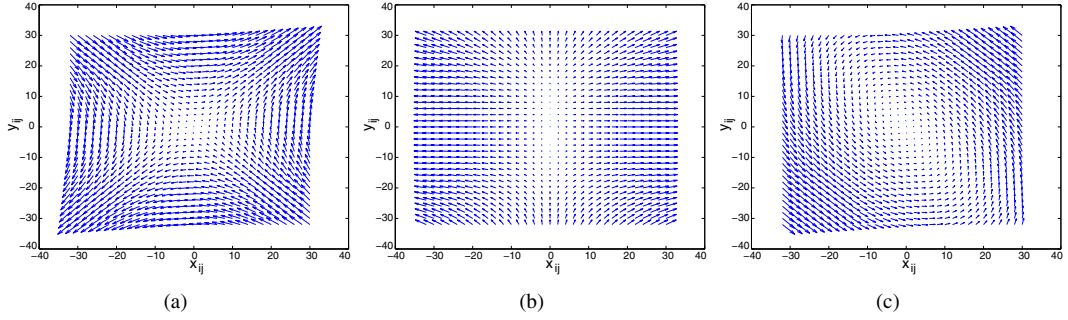


Figure 2. Synthetic flow fields (a) Diverging Flow (b) Sinusoidal Flow (c) Affine Flow.

$\{-32, -31 \dots 31\} \times \{-32, -31, \dots 31\}$ and $\mathcal{K} = 64$ as shown in figure 2. In all the three cases, we repeated the experiment 10 times for each value of σ to reduce random errors. The schematic for the computation of MSE with flow fields is shown in figure 3.

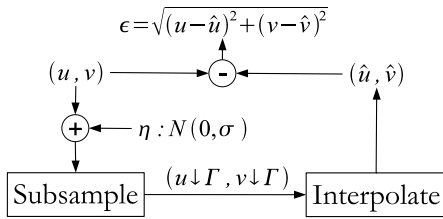


Figure 3. Error Analysis Flow

We compared two other algorithms against ours. The first algorithm was the “imresize()” function from MATLAB while the second algorithm was bilinear interpolation obtained by computing the weighted average of motion at grid locations. In both the cases, the resampling was applied to the u and v components independently. Due to limited space and the similarity of the error plots, only the errors estimated from the divergence flow are shown. The other two error plots have profiles similar to figure 4, wherein the order of accuracy of various algorithms was maintained along the same lines. From figure 4, it can be seen that the bilinear interpolation provides the ideal interpolation under low error variances, but as error variance increases, the streamline based algorithms provide more stable resampling.

3.2. Synthetic Fluid Motion

For the second set of test cases, we tested the algorithm against simulated fluid flow field¹. The simulated flow field is depicted as a “jet stream” in motion, and we tested our algorithm against a few frames. The frames

¹The synthetic fluid flow data can be obtained at <http://piv.vsj.or.jp/piv/image3d/image-e.html>

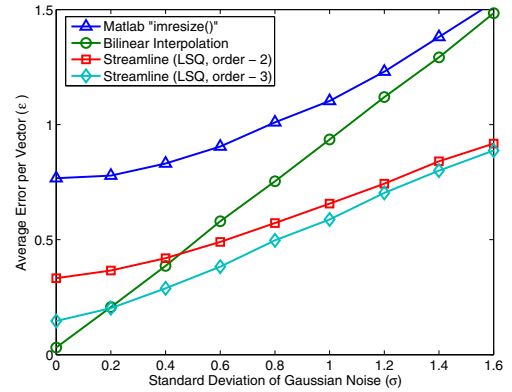


Figure 4. MSE for the diverging flow (Eq. 4) under varying noise.

that we selected for comparing the algorithms contained significant non linear dynamics as can be seen in figure 5(a). The MSE for the flow field can be observed in figure 5(b) where the streamline based algorithm produces a lower average error per vector.

3.3. Robust Estimation

One of the possible improvements that we attempted was to use RANSAC [3] for robust computation of the polynomial. RANSAC, however seemed to have some problems at higher σ . Figure 6 shows the comparison between the use of ordinary least squares (LSQ) and RANSAC. It can be seen that the MSE obtained from RANSAC is lower than the errors obtained from the LSQ solution under lower error variances. However, under higher error variances ($\sigma \geq 1.4$), RANSAC is not as effective in minimizing MSE. This counter-intuitive finding might be a result of the iterative procedure employed in RANSAC, which attempts at maximizing the number of inliers while eliminating outliers. Unfortunately as the error variance is substantially increased,

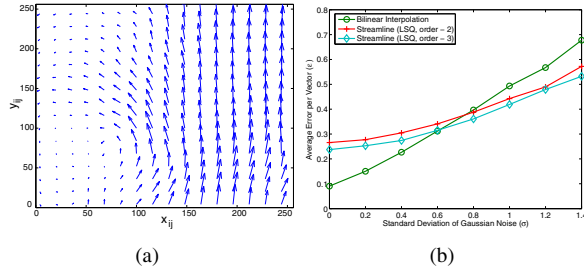


Figure 5. Synthetic Fluid Flow (a) Flow in Frame #142 (b) Average MSE.

the proportion of outliers eliminated tend to result in the higher MSE. In order to use a robust estimator that might circumvent this problem, we are in the process of incorporating an Iterative Re-weighted Least Squares estimator, which would reduce the impact of outliers on the estimation rather than eliminating them. This we hope, would successfully ameliorate the problems with the current robust estimator.

4. Discussion

The main disadvantage with our algorithm is its increased time complexity when compared to the bilinear interpolation scheme. This is because the computation of streamlines at each grid location would constitute an $\mathcal{O}(\mathcal{L}N^2)$ complexity compared to the $\mathcal{O}(N^2)$ complexity for bilinear interpolation (N^2 refers to the number of grid locations and \mathcal{L} refers to the average length of a streamline). Therefore, in a reduced noise environment, the bilinear interpolation scheme can be considered sufficient. However, in the presence of noise, the added computational time for flow-based algorithms may be warranted for increased accuracy in results.

5. Conclusion

In this paper, we describe an algorithm for resampling vector fields using local approximation of streamlines. Instead of independently interpolating the motion components, this method attempts to account for the flow characteristics while resampling. Results on parametric flow fields indicate that the algorithm achieves lower mean square error over bilinear interpolation, especially in the presence of noise. We are currently developing techniques to incorporate flow-based interpolation for discontinuous flow fields. In [7], we estimate motion at discontinuities by extending neighborhood flow characteristics using local streamlines. We intend to apply similar techniques for interpolation of

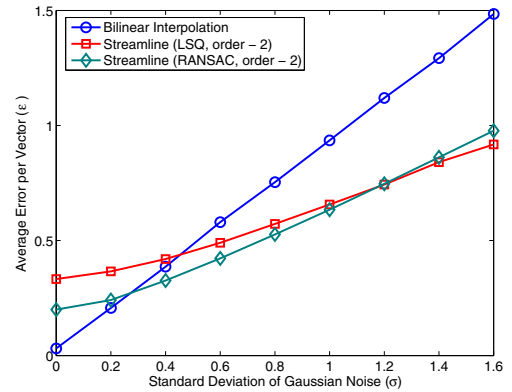


Figure 6. MSE for the divergence flow using RANSAC.

streamline coefficients at discontinuities, which is an important challenge for sea-ice motion analysis. This extension might also be useful in color image interpolation, where we intend to use our flow based interpolation on 3D color vectors.

6. Acknowledgements

This work is supported by a grant from the National Science Foundation Arctic Sciences Division [ARC-0612105 (UD)] within the Office of Polar Programs.

References

- [1] J. Barron, D. Fleet, and S. Beauchemin. Performance of optical flow techniques. *Int. J. Comp. Vision*, 12(1):43–77, February 1994.
- [2] B. Cabral and L. C. Leedom. Imaging vector fields using line integral convolution. In *Proc. Conf. on Comp. Graphics and Interactive Tech.*, pages 263–270, 1993.
- [3] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. Assoc. Comp.*, 24(6):381–395, 1981.
- [4] I.-T. R. H.263. *Video coding for low bit rate communication*. ITU-T, January 2005.
- [5] T. M. Lehmann, C. Gonner, and K. Spitzer. Survey: Interpolation methods in medical image processing. *IEEE Trans. on Medical Imaging*, 18(11):1049 – 1075, 1999.
- [6] K. Okamoto, S. Nishio, T. Kobayashi, T. Saga, and K. Takehara. Evaluation of the 3D-PIV standard images (PIV-STD project). *Journal of Visualization*, 3-2:115–124, 2000.
- [7] M. Thomas, C. A. Geiger, C. Kambhamettu, and P. Kannan. Streamline regularization for large discontinuous motion of sea ice. In *Workshop on Pattern Recognition in Remote Sensing*, December 2008.