Abstract

Eddy tracking is just finding all vectors with similar motion in its influence region and recording its moving route. Virtually a flow field is able to be replaced with a scalar dataset if its feature information can be faithfully preserved by the scalar dataset. Therefore, eddy tracking problem can be solved by tracking moving objects in an image sequence after a certain transformation. Here we propose a new method to track several moving eddies simultaneously and three main technologies are introduced: green function method, motion probability distribution and quadratic program.

1. Introduction

Eddy motion over the spatial and temporal domains is very important for the research of an unsteady flow field, because it can tell us how many moving eddies there exist and when and which eddy will appear or disappear. At present, however, the study of tracking eddy motion is still in its infancy.

In recent years there has been an increasing interest in static vector field segmentation ([1, 2]) in two-dimensional space, since vector fields are primary research objects in many scientific and technologic fields such as vector visualization and CFD. For instance, Chen et al. [1] studied 2D vector field clustering based on piecewise linear vector field approximations and an extension of Normalized Cut (Ncut)[6]. In nature the dynamic flow field segmentation and tracking is much more significant than static fields, therefore the research topic about eddy tracking will be more popular in the coming future.

2. Motion Analysis of Unsteady Flow Field

Fluid motion is revealed by the movement and variation of eddies such as expansion, shrink, rotation or advancement. Perceptually eddies can be considered as drifting on flow fields, while rotating or transforming. Hence our concerns for eddy tracking focus on two respects. One is the size of an eddy (also called influence region), which reveals its magnitude. The other is the variation of eddies. Eddy tracking of unsteady flow field is just finding all vectors with similar motion in its influence region to analyze its variation: how many moving eddies there exist now and when and which eddy will appear or disappear.

Our idea about simultaneously tracking moving eddies in unsteady flow field was originally motivated by the analogy between our purpose and tracking moving objects in an image sequence [3]. The principle of segmenting a series of images in light of motion information of objects is that pixels with same motion are grouped together and separated from the background, which is similar to eddy tracking that vectors with similar motion around an eddy should be classified as a group, and others are in other groups. We assume that eddies is drifted in a background field, while moving or transforming gradually with time, just like objects are moving in the natural image. In our work, GFM is used to build a bridge between an intensity image and a flow field first pro-

Eddy Tracking of Unsteady Flow Field

Hongyu Li
Fudan University, China
Dept. of Computer Science and Engineering
Hongyuli@citiz.net

Li Teng
Fudan University, China
Dept. of Computer Science and Engineering
tengli_hust@hotmail.com

Wenbin Chen
Fudan University, China
Dept. of Mathematics
wbchen@fudan.edu.cn

I-Fan Shen
Fudan University, China
Dept. of Computer Science and Engineering
yfshen@fudan.ac.cn

The outline of this paper is as follows. In Section 2 we briefly describe the main idea about motion analysis of unsteady flow field. Green Function Method (GFM)is introduced in section 3. Section 4 explains how to find an optimal motion segmentation, then the concept of motion probability distribution is introduced in Section 5. Section 6 presents the whole algorithm and experimental results. Conclusion is provided in Section 7.
posed in [2].

3. Green Function Method

Analyzing motion of vector fields directly is difficult for us because their features are of multiple dimensions. Consequently we need to convert complicated multidimensional problems into simple scalar problems, which means that a suitable transformation must be found to preserve feature information of vector fields faithfully.

Here GFM was chosen to implement the transformation from vector fields to intensity images. The basic principle of GFM is to find a corresponding intensity image such that its gradient or curl is able to fit the vector field best in the least square sense. In the following we will briefly discuss this method and the detailed proof about it will be presented in our future work.

According to discrete Hodge decomposition ([5, 7]), a vector field \( v \) can be decomposed into three parts:

\[
v = \nabla \phi + \nabla \times \psi + h
\]

where \( \phi \) and \( \psi \) are respectively the scalar potential and vector fields. Because \( h \) is relatively small, it can be usually ignored. Therefore, \( v \) can be represented by two scalar functions, \( \phi \) and \( \psi \), which give it 'best fit'. And these two function can be computed from \( v \) in light of the discrete convolution operation:

\[
\phi(r_i) = \sum_j \nabla G(r_j - r_i) \cdot v(r_i)
\]

\[
\psi(r_i) = \sum_j \nabla \times G(r_j - r_i) \cdot v(r_i)
\]

where \( r_i = (x_i, y_i) \) is the coordinate of \( i \)-th component of \( v \); \( v(r_i) = (v(x_i), v(y_i)) \) is the vector at position \( r_i \); \( v(x_i) \) and \( v(y_i) \) respectively correspond to the magnitude of vectors in horizontal and vertical directions and green function \( G(r_i - r_j) = -\frac{1}{\pi} e^{-\frac{||r_i - r_j||^2}{\sigma}} \) where the parameter \( \sigma \) is a ratio factor. \( \nabla \) and \( \nabla \times \) represent the gradient and curl operators respectively.

By combining \( \phi \) and \( \psi \) linearly, we can get a desired function \( \kappa \) to represent vector field \( v \):

\[
\kappa = \alpha \phi + (1 - \alpha) \psi
\]

The parameter \( \alpha \) is required to vary between zero and one. A small value of \( \alpha \) means that the divergence-free part \( \nabla \times \psi \) is more important than the curl-free one \( \nabla \phi \) in \( v \), while a large value of \( \alpha \) emphasizes the significance of \( \nabla \phi \).

Fig. 1 is an example of converting a vector field into an intensity image, where parameters \( \sigma = 0.5, \alpha = 1 \). The left is a 21 \times 21 discrete vector field with two eddies. Since it is curl-free, here \( \alpha \) is assigned one. In the intensity image (the right) obtained by GFM, each grid is a pixel corresponding to a vector and the color on it represents the magnitude of the pixel. The image, from which we can easily find eddies of its corresponding flow field, does faithfully preserve feature information of the flow field.

4. Motion Segmentation

Since motion information of objects can be computed and incorporated into the similarity between pixels, motion segmentation can be considered as a special instance of a general grouping or clustering problem ([3]). Each pixel in an image sequence can be treated as a node in an undirected weighted graph. The weight on edges connecting two nodes represents motion and spatial position similarity between these two corresponding pixels. After constructing the weighted matrix, the remaining problem is to find a best way of partitioning the graph.

4.1. Quadratic Program

The work [4] has presented and proved that a dominant set represents an optimal partition in a graph and is easily obtained by locally maximizing a quadratic program over a simplex.

For a graph \( G = (V, E) \) with a weighted matrix \( W \), the following quadratic program is considered:

\[
\begin{align*}
\text{maximize :} & \quad f(x) = x^T W x \\
\text{subject to :} & \quad x \in \{x \in \mathbb{R}^d : x \geq 0 \text{ and } e^T x = 1\}
\end{align*}
\]

The constrained condition is called standard simplex. Since a dominant set is composed of all nonzero elements of a
strict local solution vector of (2), it can be obtained by finding a strict local maximizer, and hence will correspond to an optional segmentation of $G$.

Repliactor dynamics method is very simple and efficient for solving the quadratic program. Given a non-negative $n \times n$ symmetric matrix $W$, the discrete-time replicator equation can be written as:

$$x_i(t+1) = x_i(t) \frac{(Wx(t))_i}{x(t)^TWx(t)}$$

where $x_i(t)$ is the $i$-th component of solution vector $x(t)$ after iterating $t$ times.

In our experiments of motion segmentation, the initial $x$ is guessed as $x(i) = x(i)/\sum_i x(j)$. Usually after 10 times iterations the program (2) does converge to a local strict extremum point.

Compared with Ncut method [6], the prominent advantage of solving a quadratic program is more efficient since it does not have to solve an eigenvalue system.

5. Constructing the Similarity Matrix

To segment a motion sequence, a weighted graph can be set up by treating each pixel as a node of the graph and connecting nodes in a spatial neighborhood. The weight on a graph edge connecting two pixels represents their motion similarity. To incorporate motion information into the similarity matrix, motion probability function is adopted to represent possible motion directions of pixels because of the uncertainty of their movement.

5.1. Motion Probability of Pixels

Because moving directions of pixels are unknown, motion vectors are introduced to represent possible displacement of pixels and incorporated into the similarity matrix.

In the following paragraphs, we will discuss the motion probability distribution briefly (see [6] for details). Let $l'(Q)$ denote a window centered at the pixel at location $Q \in R^2$ at time $t$. Similarity between two image patches can be computed based on the SSD difference:

$$S_i(d_Q) = \exp(-\sum_w (l'(Q_i+w) - l'^+1(Q_i+d_Q+w))^2/\sigma_{out})$$

where $w$ is within a local neighborhood of image patch $l'(Q_i)$. Because the displacement of objects between any two sequential frames is relatively small, taking size of $d_Q$ 7 × 7 is enough for our research usually. $P_i(d_Q)$ denotes the probability of an image patch at node $i$ and can be computed by:

$$P_i(d_Q) = \frac{S_i(d_Q)}{\sum_{d_Q} S_i(d_Q)}$$

An example of image patch is given in Fig. 2 with the 7 × 7 window. Fig. 2a and 2b are two frames of a series of sequential images. Fig. 2c and 2d are respectively the motion probability distributions at both pixels 1 and 2. Note they have captured the motion directions at those points as well as their associated uncertainties.

We define the distance between two image patches $i$ and $j$ as:

$$d(i,j) = 1 - \sum_{d_Q} P_i(d_Q) P_j(d_Q)$$

where $d_Q$ range over possible displacement of a node.

5.2. Similarity Matrix

The weight on a graph edge connecting two nodes represents the similarity of their motion and position. In our experiments the weight function can be defined as follows:

$$W(i,j) = \exp(-d(i,j)/\sigma_i - ||Q_i - Q_j||/\sigma_e)$$

$$W(i,j) = W(i,j)/\sqrt{G(i)G(j)}$$

where $\sigma_i$ and $\sigma_e$ are positive real constant numbers which affect the decreasing rate of $W$. (5) is the normalization of $W$, where $G(i) = \sum_k W(i,k)$. Although we assume that there exists a weight on any edge connecting two pixels, it is obvious that the weight is very small if these two pixels are too far, so their weight will be assigned zero when the spatial difference between two pixels is larger than 5. Then the weight matrix will become sparse, which can improve the efficiency of operation but does not virtually damage the quality of resulting segmentation.

6. The Complete Algorithm and Results

Our approach can be summed up as follows:

1. Using GFM, convert moving flow fields into an intensity image sequence.
2. For the image sequence, construct a weighted graph $G = (V, E)$ by treating each pixel as a node in a graph. Usually only connect nodes that are not more than 5 pixels apart in space.
3. Compute the motion probability of each node respectively and the weight matrix $W$.
4. Use discrete replicator equation (3) to iteratively find a strict local maximizer of a quadratic program (2) starting from an initial guess. With the local optimal solution, find a dominant set which is just a motion segmentation.
5. According to the corresponding relationship between flow fields and images, thus the motion segmentation and tracking of eddies can be implemented.
We have applied our method into some cases and get some desired results. Fig. 3 is an example of segmenting dynamic synthetic flow fields and tracking their eddies. Within the left three sequential vector fields there exists two moving eddies. The right is the resulting segmentation. As shown in the figure, these two moving eddies were exactly detected simultaneously and their varying process can be well understood: the left eddy has lowered gradually while the right one has heightened, but their size has no clear changes.

7. Conclusions

Eddies can expand, shrink, rotate or advance with time. Eddy tracking of unsteady flow field can help us analyze its variation: how many moving eddies there exist now and when and which eddy will appear or disappear. In this paper, we have developed a novel method to track eddy motion and the experimental results show that our approach is very efficient and desirable.

Acknowledgment

This work was supported by NSFC Grant Project named vector field segmentation, clustering and dimension reduction and the Special Funds for Major State Basic Research Projects under contract G199032804.

References


