The infinitestimal operator for the semigroup of the Frobenius-Perron operator from image sequence data: Vector fields and computational measurable dynamics from movies

N. Santitissadeekorn and E. M. Bollt

Department of Mathematics and Computer Science,Clarkson University,
Potsdam, NY 13699-5815,USA

Abstract

We present in this paper an approach to approximate the Frobenius-Perron transfer operator from a sequence of time-ordered images, that is, a movie dataset. Unlike time-series data, successive images do not provide a direct access to a trajectory of a point in a phase space; more precisely, a pixel in an image plane. Therefore, we reconstruct the velocity field from image sequences based on the infinitesimal generator of the Frobenius-Perron operator. Moreover, we relate this problem to the well-known optical flow problem from the computer vision community and we describe a validity of using the continuity equation derived from the infinitesimal operator as a constraint equation for the optical flow problem. Once the vector field, and then a discrete transfer operator are found then, in addition, we present a graph modularity method as a tool to discover basin structure in the phase space. Together with a tool to reconstruct a velocity field, this graph-based partition method provides us a way to study transport behavior and other ergodic properties of measurable dynamical systems captured only through image sequences.

1 Introduction

A development of computational methods in the burgeoning field of measurable dynamics to model and identify transport activity in both deterministic and stochastically perturbed dynamical systems, specifically through approximating the Frobenius-Perron transfer operator, demonstrate numerous applications in number of domains [1–3]. Our primary interest is to develop a numerical tool in a framework of the Frobenius-Perron transfer operator to study basin structure, transport behavior, and other ergodic properties all through processing time-series data from movies of a dynamical evolution of a probability density profile $I(x, y, t)$, for example, when observing a satellite image, one may ask about what is the underlying velocity field, and also are there any almost invariant sets. In this paper we will relate the optical flow problem to reconstruct the displacement field that transform an image to the next image to problems of measurable dynamics through the Frobenius-Perron operator.

To construct the Ulam-Galerkin approximation of the transfer operator we are required to have a knowledge of a family of transformations, see Section 2.3. However, a sequence of images does not provide a direct access to the trajectory of a point in a phase space. Successive images instead only describe an evolution in time of a distribution of brightness of image pixels. For this reason, the primary interest of this paper is to reconstruct a velocity field from a two-dimensional image sequence that transforms the intensity pattern from one image to the next image in a sequence, which is called *optical flow* computation [4]. The problem of determining the optical flow has been intensively studied in the last two decades in many areas of computer vision to perform motion detection and tracking, object segmentation, time-to-collision estimation, and etc. Recently, the analysis of the optical flow has been applied to many other science areas such as meteorology, oceanography, climatology, where image sequences from satellite platforms are the main source of their information [5–12]. However, the application of this optical flow analysis has not yet been demonstrated in the area of the measurable dynamical system, in particular, to approximate the Frobenius-Perron transfer operator from a sequential image data.

In this paper we assume that the desired velocity field is autonomous throughout the image region and so the trajectory of a point in the image region is governed by the following equation:

$$\begin{aligned}\frac{dx}{dt} &= u(x,y)\\ \frac{dy}{dt} &= v(x,y),\end{aligned} \tag{1}$$

where $u(x,y)$ and $v(x,y)$ are two unknown velocity fields to be approximated from a sequence of images, which provides to us only a temporal variation of the brightness pattern. We describe this temporal variation in a framework of the Frobenius-Perron operator. In section 2 we will show that this evolution can be expressed as the solution of the partial differential equation derived from the infinitesimal operator of the Frobenius-Perron operator [13]. This partial differential equation turns out to be the well-known continuity equation in fluid mechanics, and it will serve as the constraint equation for the optical flow computation, see Section 3. We will focus here on the application of using optical flow to generate the finite-rank approximation of the Frobenius-Perron operator instead of developing an advance algorithm to find the optical flow.

Obtaining an approximated velocity field from an image sequence permits us to approximate the Frobenius-Perron operator based on the Ulam-Galerkin method, which will be described below. The approximation will be in the matrix form called the Ulam-Galerkin matrix. Then we may generate a graph network corresponding to the Ulam-Galerkin matrix. Furthermore, we also determine the phase space partition of the dynamical system of the image sequence described by the approximated optical flow. This partition identifies the regions in the phase space that should be dynamically grouped(those in the same basin). In the view point of the graph theory, we must discover the *community structure* of the graph-the partition that groups together those nodes within which they are densely connected, but their connection to other nodes are comparatively sparse [14]. The approach to discover the community structure of the network used in this paper is called the *modularity* method [14–17] based on the optimization of the modularity measure of the network. Note that in a different approach, recent efforts have focused to identify the number and location of *almost invariant* sets; those subsets of a state space where trajectories tend to stay for comparatively long periods of time before they leave into other regions, in a context of the *congestion* of a graph based on a multi-commodity flow on the graph [18–21]. In this sense, the graph community structure and the almost invariant sets are essentially the same tool to approximate the basin structure of the phase space.

2 Background

We review in this section some properties of the Frobenius-Perron operator that will help us to relate the inverse problem of the optical flow computation, which will be reviewed in Section 3, to the approximation of the Frobenius-Perron transfer operator.

2.1 Frobenius-Perron and Koopman operators

Let $(\mathbb{X}, \mathbb{A}, \mu)$ be a measure space. Let $F : \mathbb{X} \to \mathbb{X}$ be a non-singular measurable transformation on $(\mathbb{X}, \mathbb{A}, \mu)$,that is,

$$\mu(F^{-1}(A)) = 0 \qquad \text{for each } A \in \mathbb{A} \text{ such that } \mu(A) = 0. \tag{2}$$

The *Frobenius Perron operator*, $P : L^1(\mathbb{X}) \to L^1(\mathbb{X})$ with respect to F is defined by [13],

$$Pf(x) = \int_{\mathbb{X}} \delta(x - F(y))f(y)dy, \tag{3}$$

where $f(x)$ is a probability density function (PDF) defined in $L^1(\mathbb{X})$. Thus $Pf(x)$ gives us a new probability density function, which is unique a.e., and depend on the discrete time transformation F and the probability density function $f(x)$. For all measurable sets $A \subset \mathbb{A}$ the Frobenius-Perron operator satisfies the discrete-time continuity equation [13]

$$\int_{F^{-1}(A)} f(x)dx = \int_A Pf(x)dx, \qquad \text{for each } A \in \mathbb{A} \tag{4}$$

The operator $K : L^\infty \to L^\infty$, called the *Koopman Operator* with respect to F, is defined by

$$Kg(x) = g(F(x)) \tag{5}$$

for $g \in L^\infty$. For us, the key property of the Koopman operator is that it is adjoint to the Frobenius-Perron operator. That is for every $\rho \in L^1, g \in L^\infty$,

$$\langle P\rho, g\rangle = \langle \rho, Kg\rangle, \tag{6}$$

where we denote the bilinear form $\langle \cdot, \cdot \rangle_{L^1(\mathbb{X})\times L^\infty(\mathbb{X})}$ by $\langle \cdot, \cdot \rangle$ throughout this paper.

2.2 Infinitesimal generators

In this section we review some background of the infinitesimal generator of the semigroup of the Frobenius-Perron operator, of which the details can be found in Lasota and Mackey [13]. The key result in this section will form the main constraint equation used in the optical flow computation.

Consider a d-dimensional system of ordinary differential equations

$$\frac{dx_i}{dt} = F_i(x), \qquad i = 1, \ldots, d, \tag{7}$$

where $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$. This system gives a continuous time process $\{S_t\}_{t\geq 0}$ defined by

$$S_t(x^0) = x(t), \tag{8}$$

where $x(t)$ is the solution of Eq. (7) with the initial condition $x^0 = x(0)$. Assuming that existence and uniqueness of solutions of Eq. (7) are satisfied, one can show that the family $\{S_t\}_{t\geq 0}$ forms a continuous semigroup of transformations corresponding to Eq.(7). Additionally, we can also defined the Frobenius-Perron operator and the Koopman operator for a continuous time process associated to the semigroup of transformation $\{S_t\}_{t\geq 0}$ in the same fashion of a discrete time process as previously described, that is,

$$\int_{S_t^{-1}(A)} f(x)d\mu = \int_A P_t f(x) d\mu \qquad \text{for each} A \in \mathbb{A}, \tag{9}$$

and

$$K_t g(x^0) = g(S_t(x^0)). \tag{10}$$

Now we can state the aim of this section more precisely. That is we want to discuss about the evolution of time-dependent density function $I(t, x) \equiv P_t f(x)$ for some initial density function $f(x)$ in $L^1(\mathbb{X})$. This can be developed by using the infinitesimal operator as discussed in Lasota and Mackey [13]. We will briefly repeat this concept here for a sake of completeness, and we will emphasize how we will relate it to the optical flow computation.

For a semigroup of contractions $\{T_t\}_{t\geq 0}$ we define by $D(A)$ the set of all $f(x) \in L^p(\mathbb{X}), 1 \leq p \leq \infty$, such that the limit

$$Af = \lim_{t\to 0} \frac{T_t f - f}{t}, \tag{11}$$

exists in the sense of strong convergence, that is,

$$\lim_{t\to 0} \left\| Af - \frac{T_t f - f}{t} \right\|_L = 0. \tag{12}$$

The operator $A : D(A) \to L$ is called the *infinitesimal generator*. Let $I(t) \equiv I(t,x) = T_t f(x)$ for fixed $f(x) \in D(A)$. The function $I'(t) \equiv I'(t)(x) \in L^p(\mathbb{X})$ is said to be the *strong derivative* of $I(t)$ if it satisfies the following condition:

$$\lim_{t\to 0} \left\| I'(t) - \frac{I(t) - f(x)}{t} \right\|_{L^p} = 0. \tag{13}$$

In this sense, $I'(t)$ describes the derivative of the ensemble of points with respect to time t. It is shown in [13] that for $t \geq 0$ and $I(t,x) \in D(A)$ we have that $I'(t)$ exists and satisfies the equation

$$I'(t) = AI(t) \tag{14}$$

with the initial condition $I(0) = f(x)$.

We are now in a position to discuss about the infinitesimal generator of the Frobenius-Perron operator generated by the family of transformation $\{S_t\}_{t\geq 0}$ and the evolution of time-dependent density function $I(t,x)$ under an action of the Frobenius-Perron operator. This will be done indirectly through the adjoint property of the Forbenius-Perron and Koopman operators.

It follows directly from the definition of the Koopman operator Eq. 10 that the infinitesimal of the Koopman operator A_K is

$$A_K g(x) = \lim_{t\to 0} \frac{g(S_t(x^0)) - g(x^0)}{t} = \frac{g(x(t)) - g(x^0)}{t}. \tag{15}$$

If g is continuously differentiable with compact support, we can apply the mean value theorem to obtain

$$A_K g(x) = \sum_{i=1}^{d} \frac{\partial g}{\partial x_i} F_i(x). \tag{16}$$

Combining equations (14) and (16) we conclude that the function

$$I(t,x) = K_t f(x) \tag{17}$$

satisfies the first-order partial differential equation

$$\frac{\partial I}{\partial t} - \sum_{i=1}^{d} \frac{\partial I}{\partial x_i} F_i(x) = 0. \tag{18}$$

Now we are able to discuss about derivation of the infinitesimal generator for the semigroup of Frobenius-Perron operators generated by the family $\{S_t\}_{t\geq 0}$ associated the ode given by Eq. (7).

Let $f \in D(A_{FP})$ and $g \in D(A_K)$, where A_{FP} and A_K denote the infinitesimal operators of the semigroups of the Frobenius-Perron and Koopman operators, respectively. Using the adjoint property of the two operators it can be shown that

$$\langle (P_t f - f)/t, g \rangle = \langle f, (K_t g - g)/t \rangle. \tag{19}$$

Taking the limit as $t \rightarrow 0$ we obtain

$$\langle A_{FP}, g \rangle = \langle f, A_k g \rangle. \tag{20}$$

Provided that g and f are continuously differentiable and g has compact support we can show that [13]

$$\langle A_{FP}, g \rangle = \langle -\sum_{i=1}^{d} \frac{\partial f F_i}{\partial x_i}, g \rangle. \tag{21}$$

Hence, we can conclude that

$$A_{FP} = -\sum_{i=1}^{d} \frac{\partial f F_i}{\partial x_i} = 0. \tag{22}$$

Again, we can use equations (14) and (22) to conclude that the function

$$I(t, x) = P_t f(x) \tag{23}$$

satisfies the partial differential equation (continuity equation)

$$\frac{\partial I}{\partial t} + \sum_{i=1}^{d} \frac{\partial I F_i}{\partial x_i} = 0. \tag{24}$$

Note that this equation is actually the same as the well-known continuity equation in fluid mechanics, but now it is a statement of conservation of density function of ensembles of trajectories. This equation will play an important role as a constraint equation to the approximation of the optical flow from an image sequence. Note also that we are using $I(t, x, y)$ to denote the continuous time density profile, which we have denoted I in deference and relating this measurable dynamic notion to the usual notation for similar ideas in image processing using optical flow technique. Comparing the two approaches will help guide us toward a robust method for our problem in measurable dynamics.

Let us summarize here an application of materials presented in this section to our problem of the optical flow computation. First, recall that we view a sequence of two-dimensional images as a family of time-dependent density functions $I(t, x, y)$ and we assume that these density functions evolve in time under the action of the Frobenius-Perron operator associated to the continuous semigroup of transformation $\{S_t\}_{t\geq 0}$ corresponding to Eq. (7). Note that the continuous time-dependent density relates to the discrete time density by sampling at $t \in \mathbb{Z}$. Therefore, base on this idea, we can relate evolution of time-dependent images $I(t, x, y)$ to the velocity field, which we want to reconstruct, via Eq. (24). For this particular case, it can be rewritten by,

$$\frac{\partial I}{\partial t} + div(I\mathbf{v}) = 0, \tag{25}$$

where $\mathbf{v} = [u(x, y), v(x, y)]$ is the unknown velocity field to be reconstructed from a sequence of images. Therefore, we have to solve an inverse problem of the Frobenius-Perron operator. This problem is clearly an ill-posed problem since we need to solve for two unknown velocity fields from Eq. (25) alone. We defer our discussion of this problem to later section.

In the subsequent section we will discuss about the finite-rank approximation of the Frobenius-Perron operator. Note that our main interest is to construct this matrix approximation of the Frobenius-Perron operator, called the *Ulam-Galerkin* matrix, from an image sequence and to identifies the almost invariant set corresponding to this matrix. However, in order to achieve the Ulam-Galerkin matrix, we must first reconstruct the underlying velocity fields from an image sequence, which subsequently allows us to estimate the semigroup $\{S_t\}_{t\geq 0}$ corresponding to Eq. (7). We will discuss about this idea in later section.

2.3 Finite-rank approximation

We use the Ulam-Galerkin method, which is a particular case of Galerkin's method [22, 23], to approximate the Frobenius-Perron operator. We use the projection of the infinite dimensional linear space $L^1(\mathbb{X})$ with basis functions $\{\phi_i(x)_{i=1}^{\infty}\} \subset L^1(\mathbb{X})$ on to a finite dimensional linear subspace with a subset of the basis functions,

$$\triangle_N = \text{span}\{\phi_i(x)\}_{i=1}^{N}. \tag{26}$$

For the Galerkin's method this projection,

$$\Pi : L^1(\mathbb{X}) \rightarrow \triangle_N, \tag{27}$$

maps an operator from the infinite-dimensional space to an operator of finite rank $N \times N$ matrix by using the inner product

$$A_{i,j} = \langle P\phi_i, \phi_j \rangle = \int_{\mathbb{X}} P\phi_i(x)\phi_j(x)dx. \tag{28}$$

The quality of this approximation is discussed in many excellent references including [24–28]. For the Ulam's method [22] the basis functions are a family of characteristic functions

$$\phi_i(x) = \chi_{B_i}(x) = 1 \text{ for } x \in B_i \text{ and zero otherwise.} \tag{29}$$

By using Eq. (28) the matrix approximation of the Frobenius-Perron operator has the form of

$$A_{i,j} = \frac{m(B_i \cap F^{-1}(B_j))}{m(B_i)}. \tag{30}$$

where m denotes the Lebesgue measure on $\mathbb{X}$ and $\{B_i\}_{i=1}^N$ is a family of boxes or triangles of the partition that covers $\mathbb{X}$ and indexed in terms of nested refinements [22]. This $A_{i,j}$ can be interpreted as the ratio of the fraction of the box B_i that will be mapped inside the box B_j after an application of a map to the measure of B_i. Note that if we only have a test orbit $\{x_j\}_{j=1}^N$, which is actually the main interest of this paper, the Lebesgue measure can be approximated by a counting measure λ and the matrix approximation of the Frobenius-Perron operator becomes

$$A_{i,j} = \frac{\lambda(\{x_k | x_k \in B_i \text{ and } F(x_k) \in B_j\})}{\lambda(\{x_k \in B_i\})}. \tag{31}$$

3 Optical flow

The main body of this section describes an approach to extract the velocity field that transforms an intensity pattern of one image into the next image in a sequence. This problem is referred to as the *optical flow* problem in image processing. Based on the framework of the Frobenius-Perron operator reviewed in the last section, this problem is inherently an inverse problem and it is also ill-posed. There are numerous methods to the optical flow problem which can be developed to emphasize different aspects of the expected solution. We will review some of these below in order to better understand how to apply optical flow methods to measurable dynamics.

3.1 Optical flow constraint

As previously mentioned, we assume that the flow of an image pixel can be described by a two-dimensional system of ordinary differential equations

$$\begin{aligned} \frac{dx}{dt} &= u(x,y) \\ \frac{dy}{dt} &= v(x,y), \end{aligned} \tag{32}$$

where $u(x,y)$ and $v(x,y)$ are two unknown velocity fields to be approximated from a sequence of images. Let $I(x,y,t)$ represent a gray-scale intensity function of an image, a function of brightness of a pixel at point (x,y) and (discrete) time t. Since the aim of this work is to approximate the Frobenius-Perron operator from a sequence of images, the optical flow constraint for this matter, which will be called the *Frobenius-Perron constraint* (FPC), is the continuity equation (24) derived in the previous section:

$$\frac{\partial I}{\partial t} + div(I\mathbf{v}) = 0 \tag{33}$$

where $\mathbf{v} = [u(x,y), v(x,y)]$ is the unknown velocity field. Recall that this equation is derived from the infinitesimal generator of the Frobenius-Perron operator [13], and so it describes the temporal variation of the distribution of brightness under $[u(x,y), v(x,y)]$. In the language of fluid mechanics, this equation also describes the temporal variation of the brightness within an infinitesimal volume as evolved by the flux of the brightness through the boundary surface of the volume. As a special case of this constraint when $div(\mathbf{v})$ is zero throughout the image plane, we arrive at the classical *brightness constancy constraint*, which is popularized by the Horn and Schunck formulation of optical flow [4],

$$\frac{dI}{dt} = \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t} = 0. \tag{34}$$

The above equation assumes that the intensity pattern of local-time varying image regions are constant under a motion in a short time duration, which follows from the first-order Taylor series expansion about $I(x,y,t)$ and Eq. (34):

$$I(x+\delta x, y+\delta y, t+\delta t) = I(x,y,t) + \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t + O^2. \tag{35}$$

The challenge is that the optical flow computation based only either on the constraint equation (33) or (34) is an ill-posed problem because on each location and each time, we have to solve a single scalar equation for two scalar unknowns u and v. In the case of the brightness constancy assumption, this is called the *aperture problem*, where only the normal component of the velocity field, given in the direction of the gradient $\nabla \mathbf{v}$ can be solved from the constraint equation (34), but not the tangential component. Numerous methods have been proposed to overcome this ill-posed problem, which can be categorized into two general classifications. The *local* methods such as the Lucas-Kanade method [29] and the structure of tensor method [30] employ the optimization of some local energy-like expression, whereas the *global* approaches attempt to minimize a global energy functional [4]. Survey and comparison of various methods was demonstrated by Barron [31], and Galvin [32].

We resort to the Tikhonov regularization technique [33], which belongs to the class of global methods, to cope with the ill-posed problem. The idea is to approximate the solution of the constraint equation by solving a minimization problem of the form

$$\inf_{\mathbf{v}} \int_{\Omega} (F(\mathbf{v}) + S(\mathbf{v})) d\Omega, \tag{36}$$

where $F(\mathbf{v})$ is the data fidelity term based either on the constraint equation (33) or (34) and $S(\mathbf{v})$ is an additional regularization term to stabilize the solution and to relax the constraint used in the data fidelity term. The classical regularization term proposed by Horn and Schunck [4] is the so-called *smoothness* constraint:

$$S(\mathbf{v}) = \alpha(\|\nabla u(x, y)\|^2 + \|\nabla v(x, y)\|^2), \tag{37}$$

where α is a constant and the norm is the standard L^2 norm. This constraint claims that neighboring pixels of a point in a sequence of images are likely to move in a similar way, i.e. the motion vectors are spatially varying in a smooth way. Thus it encourages the isotropic smoothness of the recovered optical flow without taking into account the discontinuities at the edges where the gradient of intensity is large. Another alternative to the smoothness regularizer is the so-called div-curl regularizer [34]

$$S(\mathbf{v}) = \alpha \|div \mathbf{v}\|^2 + \beta \|curl \mathbf{v}\|^2, \tag{38}$$

which reduces to the smoothness constraint when $\alpha = \beta$.

In this paper, we consider the case that image pixels in consecutive images do not present a very strong loss or gain of luminance due to a large divergence of velocity field, which is true in a video sequence with a high frame rate, and so the classical brightness constancy equation is our preferable choice. However, it should be noted that in satellite images of motion of cloud or a sequence of fluid imagery, cloud or fluid may exhibit a high temporal deformation and large displacement between two

Table 1: Matched pairs of pre-filter(p) and derivative kernels(d_i) of the i^{th} order of various sizes.

3-tap	p: 0.2298 0.5402 0.2298
	d1: 0.4252 0.0000 -0.4252
	d2: 0.3557 -0.7114 0.3557
5-tap	p: 0.0376 0.2491 0.4263 0.2491 0.0376
	d1: 0.1096 0.2766 0.0000 -0.2766 -0.1096
	d2: 0.2190 -0.0007 -0.4366 -0.0007 0.2190
7-tap	p: 0.0047 0.0693 0.2454 0.3611 0.2454 0.0693 0.0047
	d1: 0.0187 0.1253 0.1930 0.0000 -0.1930 -0.1253 -0.0187
	d2: 0.0543 0.1370 -0.0534 -0.2758 -0.0534 0.1370 0.0543

consecutive frames, and the brightness constancy cannot be nearly satisfied. In such cases, the development of the minimization technique embedded by the FPC Eq. (24) becomes more attractive [5, 6, 12]. Thus, in this paper we will solve the minimization problem Eq. (36) using the brightness constancy constraint Eq. (34) for the data fidelity term and the div-curl regularizer Eq. (38) for the regularization term:

$$\inf_{\mathbf{v}} \int_{\Omega} (I_x u + I_y v + I_t) d\Omega + \int_{\Omega} \alpha \|\nabla div\mathbf{v}\|^2 + \beta \|\nabla curl\mathbf{v}\|^2 d\Omega, \tag{39}$$

The associate Euler-Lagrange equation for minimization of the problem Eq. (39) is the following pair of PDEs:

$$\begin{aligned} \alpha u_{xx} + \beta u_{yy} + (\alpha - \beta) v_{xy} &= I_x(I_x u + I_y v + I_t) \\ \beta v_{xx} + \alpha v_{yy} + (\alpha - \beta) u_{xy} &= I_y(I_x u + I_y v + I_t). \end{aligned} \tag{40}$$

3.2 Implementation

Numerical solution of the PDE Eq. (40) is presented in this section. First, regardless of the technique to implement Eq. (40), appropriate evaluation of image derivatives is required. We use the Simoncelli's matched filter method [35] that was demonstrated to have superior accuracy to the traditional central and backward differences. Simoncelli formulated a set of matched pair of derivative filters and lower pass pre-filters as a filter design problem. Below are pairs of matched pre-filters and derivative kernels of various sizes. To obtain I_x we convolve the n-tap smoothing kernel in the time dimension t,which combines n images into one image,and then convolves again the resultant image with the smoothing kernel in the y dimension, and then convolves that result with the differentiation kernel to obtain the final result, I_x. This procedure can be summarized in the following equation:

$$I_x(x, y, t) \approx d1 * p^T * (p(0)I(x, y, t - \frac{n-1}{2}) + \ldots + p(n)I(x, y, t + \frac{n-1}{2})). \tag{41}$$

The computation of I_y and I_t can be done in the same fashion.

Let us now discuss an algorithm to compute the solution of the PDE Eq. (40). We denote by u_{ij} the approximation of u at the pixel (i, j). A numerical solution of the Euler-Lagrange equation Eq. (40) via a finite different method resorting the Simoncelli's derivative kernels takes the following forms

$$\begin{aligned} (\alpha+\beta+I_x^2)u_{ij} + I_x I_y v_{ij} &= (\alpha+\beta)\overline{u}_{ij} + (\alpha-\beta)\delta_{xy} v_{ij} - I_x I_t \\ I_x I_y u_{ij} + (\alpha+\beta+I_y^2)v_{ij} &= (\alpha+\beta)\overline{v}_{ij} + (\alpha-\beta)\delta_{xy} u_{ij} - I_x I_t, \end{aligned} \tag{42}$$

where $\overline{u}_{ij} = (\alpha\overline{u}^x_{ij} + \beta\overline{u}^y_{ij})/(\alpha+\beta)$, $\overline{u}^x_{ij} = (u * w_{xx})_{ij}$, $w_{xx} = d_2 * p^T$ but setting the element $(n/2+1/2, n/2+1/2)$ to zero, $\overline{u}^y_{ij} = (u * w_{yy})_{ij}$, $w_{yy} = d_2^T * p$ but setting the element $(n/2+1/2, n/2+1/2)$ to zero, and $\delta_{xy} u_{ij} = (u * w_{xy})_{ij}$ for $w_{xy} = d_1 * p^T * d_1^T * p$.

Thus we may estimate $[u_{ij}, v_{ij}]$ iteratively by

$$\begin{aligned} u^{n+1}_{ij} &= \overline{u}^n_{ij} - \frac{I_x(I_x\overline{u}^n_{ij} + I_y\overline{v}^n_{ij} + I_t) + L_1}{\alpha+\beta+I_x^2+I_y^2} \\ v^{n+1}_{ij} &= \overline{v}^n_{ij} - \frac{I_y(I_x\overline{u}^n_{ij} + I_y\overline{v}^n_{ij} + I_t) + L_2}{\alpha+\beta+I_x^2+I_y^2}, \end{aligned} \tag{43}$$

where L_1 and L_2 are given by

$$\begin{aligned} L_1 &= \frac{\alpha-\beta}{\alpha+\beta}\left[(\alpha+\beta+I_y^2)\delta_{xy} v^n_{ij} - I_x I_y \delta_{xy} u^n_{ij}\right] \\ L_2 &= \frac{\alpha-\beta}{\alpha+\beta}\left[(\alpha+\beta+I_x^2)\delta_{xy} u^n_{ij} - I_x I_y \delta_{xy} v^n_{ij}\right]. \end{aligned} \tag{44}$$

4 Graph modularity method

Here we present a computational method for discovering almost invariant regions for understanding basin structures and basin leakage. After obtaining the optical flow we may use it as a velocity field to transform randomly chosen points on a grid partitioning of the phase space using Eq. (32)so that we can generate the Ulam-Galerkin matrix, a finite-rank approximation of Frobenius-Perron operator. Note that since we have merely a sequence of images to start with and lack of the knowledge of the physical domain of the phase space, we can just choose the doain of the phase space arbitrarily. As such we choose our domain to be the unit box $[0, 1] \times [0, 1]$, and we select random points in this domain. Next, we generate a graph network corresponding to the Ulam-Galerkin matrix by solving Eq. (32) using those randomly selected points in the grid partitioning of the unit box as the initial points. The nodes in the graph represent the grids used to partition the phase space and the edges describe the transport between each grid. However, to obtain a meaningful graph we need to determine the time duration of the trajectory of the initial points. If the time duration is too short, some initial points may remain in the same grid in which they initially reside, see Figure 1. This circumstance induces some self-connected edges in the graph network that do not provide useful information for a graph partition method. Nevertheless, a self-connected edge generated by a grid that contains a fix point cannot be avoided. A useful criteria to choose the time duration can be formulated using the well-known Gronwall's inequality [36].

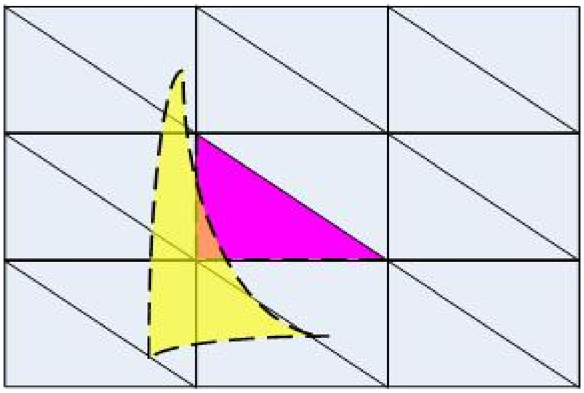


Figure 1: When solving Equation (32) the time duration has to be chosen in a way that the (triangular) grid is expanded far enough so that it lie across itself as less as possible. Then the solution of the majority of randomly chosen points will not remain inside the initial grid. This prevents us from generating a graph network with inapplicable self-connected edges.

Now our purpose is to discover community structure in the graph network to identify the existing basin structure in the phase space. Therefore, our point of view is to map the problem of phase space partition of the dynamical system captured through a sequence of image into a problem of partitioning a graph network representing the action of the transfer operator. Note that although points in the phase space may not travel across from one basin to another, it is possible to have edges in the graph that connect between each community; this represents a basin in the phase space, due to the grids that lie across the basin boundary as seen in Figure 2. However, this is only a small boundary effect. Likewise, there can be a small amount of spurious basin leakage due to finite grid effects.

Many methods have been proposed in recent years [14–16, 37] to appropriately partition a graph network. Loosely, community structure is a partition of the network into subgraphs such that there are relatively more connections within each defined component than between components, a sort of self clustering. We again note that the

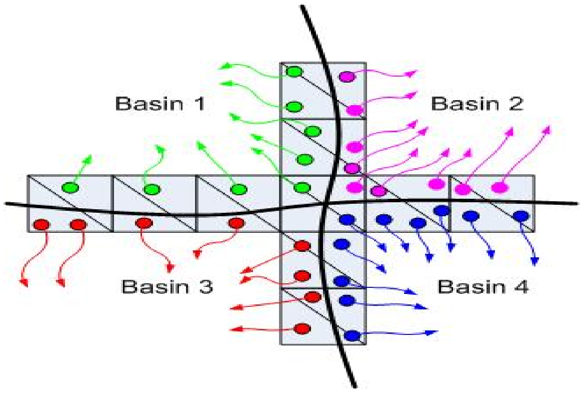


Figure 2: The phase space in this figure are partitioned by triangular grids. The edges that connect the nodes corresponding to the triangles that lie across the basin boundary are the intergroup edges, which connect between each community in the graph network.

problem of identifying the number and location of almost invariant sets is equivalent to that of discovering the graph community structure, and an approach for such a problem has been recently proposed in [18–21] in a context of the congestion of a graph based on a multi-commodity flow on the graph. In this paper, we employ a new and efficient algorithm proposed by Newman [16] for detecting community structure based on the optimization of the "*modularity*" measure.

First we discuss the concept of the modularity as a measure of quality of a proposed community partition. The modularity is a cost function associated with a partitioning of a given graph G,

$$Q : \mathcal{P}_G \to \Re, \tag{45}$$

where $\mathcal{P}_G$ is the set of all sub partitions P, of a given graph G. Given a graph G and a (test) partition, $P \in \mathcal{P}_G$, $\mathrm{P} = \cup_k \mathrm{P}_k$, each P is a set of subsets P_k, and P_k is a collection of vertices of G. P includes all of the vertices of G. We will refer to P_k as "community"-k. The modularity of the partition P is meant to reflect the quality of the split into self clustered elements P_k A modularity measure is defined by,

$$Q(\mathrm{P}) = \sum (e_{ii} - a_i^2), \tag{46}$$

where e_{ij} is the fraction of edges that connect vertices in community i to those in community j and,

$$a_i = \sum e_{ij}, \tag{47}$$

represents the fraction of edges that connects to community i. Thus $Q(\mathrm{P})$ measures the difference between the fraction of the within-community edges and the expectation of the same quantity in the network with same community partition created by randomizing all connections between vertices. Therefore, $Q(\mathrm{P})$ approaches 0 for a randomly connected network, and approaches $Q(\mathrm{P}) = 1$, if the network has a strong community structure. We may then optimize Q over all possible partitions to discover the best community structure

$$Q = \max_{\mathrm{P} \in \mathcal{P}_G} Q(\mathrm{P}). \tag{48}$$

The true optimization is, however, very costly to implement in practice for very large networks as an NP problem. Clauset, Newman, and Moore [17] proposed an approximate optimization algorithm based on greedy optimization. Suppose that we have a graph with n vertices and m edges. This algorithm starts with each vertex being the only member of one of n communities. At each step, the change in $Q(\mathrm{P})$ is computed after joining a pair of communities together and then choosing the pair that gives the greatest increase or smallest decrease in $Q(\mathrm{P})$. The algorithm stops after $n-1$ such joins in which a single community is left. This algorithm therefore runs in time $O((m+n)n)$, or $O(n^2)$ on a sparse graph ($n \sim m$). Note that using more sophisticated data structure as introduced in [17] can reduce a run time to $O(n \log^2 n)$.

5 Results

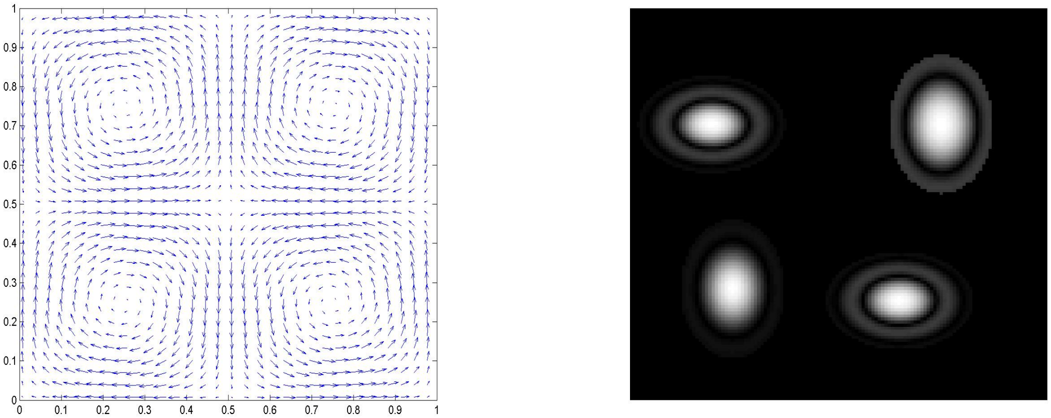
5.1 Example 1

Consider the following differential equation as the first example:

$$\begin{aligned} \frac{dx}{dt} &= -sin(2\pi x)cos(2\pi y) \\ \frac{dy}{dt} &= cos(2\pi x)sin(2\pi y). \end{aligned} \tag{49}$$

Now we may use the flow continuity equation (24) to numerically translate an initial distribution in time to generate a synthetic movie data set $I(x, y, t)$ representing a time evolving spatial density function. Figures 3(a) and 3(b) illustrate, respectively, the velocity field of this system and a given initial distribution function and Figures 4(a)- 4(d) show a sequence of some images that are captured from our numerical simulation of Eq. (24).

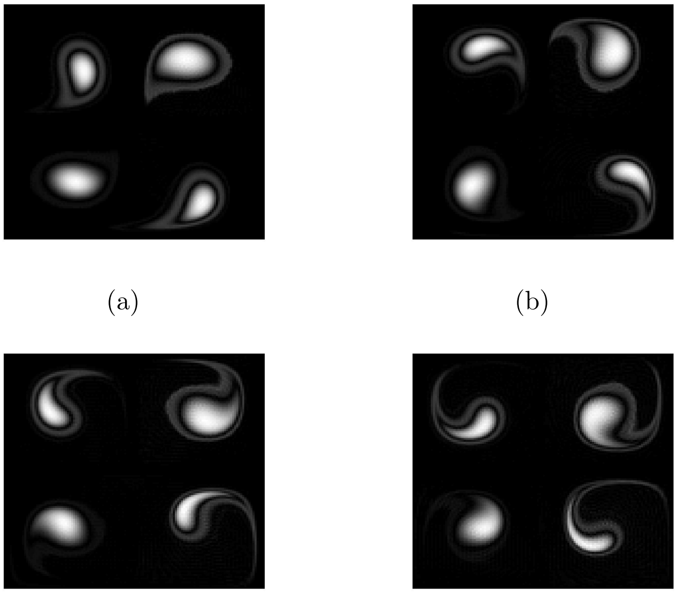
Based on this sequence of images we benchmark our approach by comparing the approximated optical flow field extracted from the images by the regularization method explained in Section 3 and the exact velocity field in Equation (49). First, let us investigate the sensitivity of the method with respect to variation of the parameters α and β defined in Eq. (38). We quantitatively evaluate our result via the



(a)

(b)

Figure 3: (a) The velocity field of the dynamical system (49). It can be seen that there consists of four basins separated by two nullclines $x = 0.5$ and $y = 0.5$. (b) The initial distribution at time $t = 0$.



(a)

(b)

(c)

(d)

Figure 4: A sequence of some images captured from the numerical simulation of the flow continuity equation (24) with the velocity field given by the ode (49).

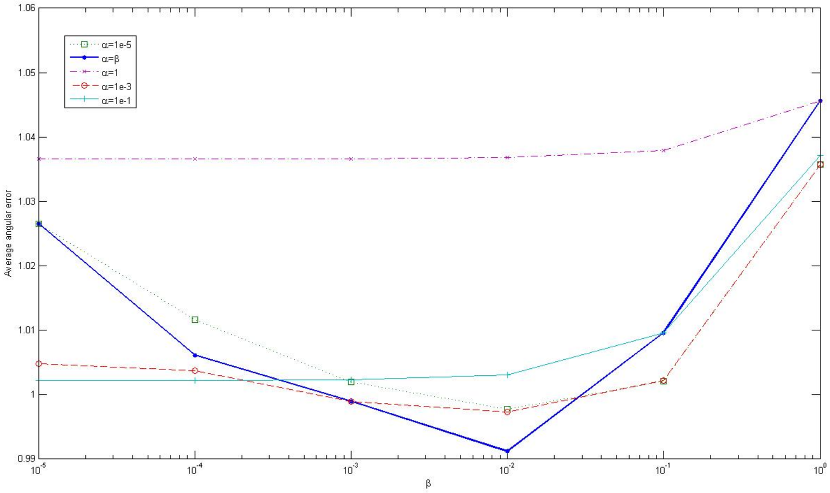
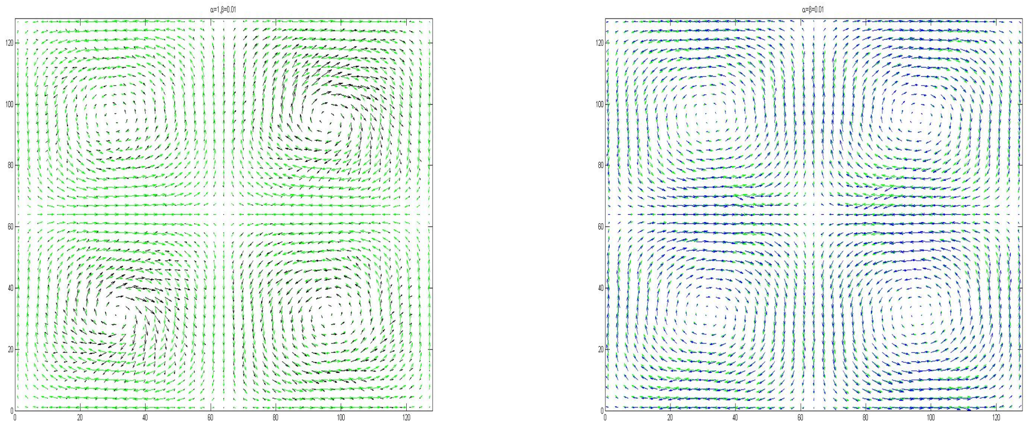


Figure 5: A plot of the average angular error with respect to parameters α and β. The thick line represents the case of the smoothness constraint (37) where $\alpha = \beta$. Although it is not conclusive that the smoothness constraint performs better than the div-curl regularizer in this experiment, our experimental result demonstrates that the optimal value is obtained when using the smoothness regularizer with $\alpha = \beta = 0.01$

average angular error:

$$e_{ang} = \arccos\left(\frac{u_{ex}u_e + v_{ex}v_e + 1}{\sqrt{(u_{ex}^2 + v_{ex}^2 + 1)(u_e^2 + v_e^2 + 1)}}\right), \tag{50}$$

where (u_{ex}, v_{ex}) denotes the exact velocity field and $(u_e v_e)$ is the estimated velocity. Remark that we do not evaluate the error via a norm because in general we would lack information about the physical domain of the image plane and we can only recover the velocity field whose dynamics is in some representative of the true dynamical system, perhaps in the sense of "almost conjugacy" [38]. Observe that for the images sequence in this example the smoothness regularizer performs better than the div-curl regularizer at the experimentally optimal value $\beta = 0.01$, see Figure 5. Nevertheless, the div-curl regularizer produces better results for other value of β. To evaluate the results qualitatively we lay the exact velocity field on the estimated velocity fields for $\alpha = \beta = 0.01$ and $\alpha = 1, \beta = 0.01$ as illustrated in Figure 6. We notice that the div-curl regularizer does not isotropically smooth the velocity field; hence we obtain only a velocity field where the motion take places meaning that there is a nontrivial support of the dataset $I(x, y, t)$. However, the smoothness regularizer produces the velocity field throughout the whole image region, notwithstanding an absence of motion in some regions, see also Figure 7. Moreover, we observe that the approximated velocity field from the smoothness constraint maintains qualitative qualities of the exact velocity fields. Therefore, we want to investigate the topological structure of the result. Note that the phase portrait of the exact velocity field has four centers with eigenvalues $\pm 2\pi i$ at points $\{(0.25, 0.25), (0.25, 0.75), (0.75, 0.25), (0.75, 0.75)\}$and a saddle point with eigenvalue $\pm 2\pi$ at the central point $(0.5, 0.5)$ of the phase space. We compute the eigenvalue of the points in the image region corresponding to those points for the velocity field approximated from the smoothness constraint with $\alpha = 0.01$ and plot the result in Figure 8. The result shows that we recover the saddle node at the central point of the image region, but we obtain four spiral sinks at those points instead of center nodes; however, the real part of their eigenvalues are much smaller than the imaginary parts, which indicates that they are suggestive of a type of center node.

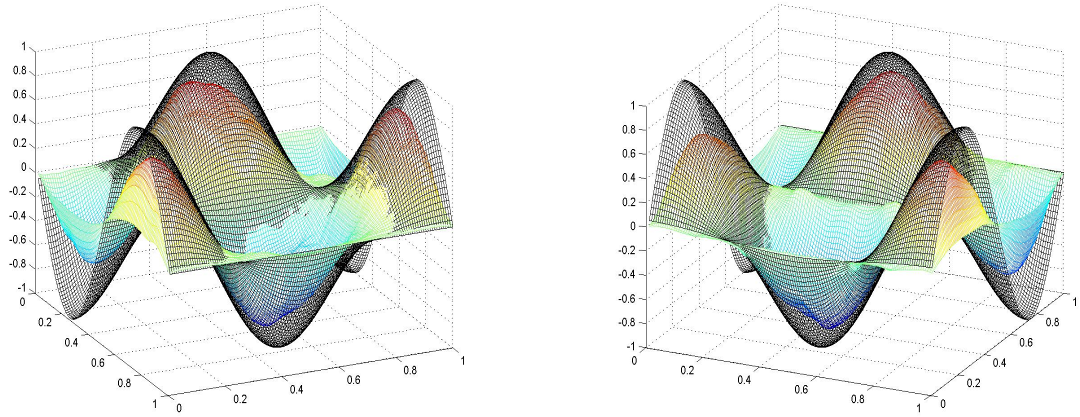


(a)

(b)

Figure 6: (a) Exact velocity field(in green arrows) and the optical flow field for $\alpha = 1, \beta = 0.01$ shown in black arrows. (b) Exact velocity field(in green arrows) and the optical flow field for $\alpha = \beta = 0.01$ shown in black arrows.

Now let us discuss how to apply the optical flow result to approximate the Ulam-Galerkin approximation of the Frobenius-Perron transfer operator. We use the approximated velocity field to advance randomly distributed points in time to generate a finite-rank approximation of the Frobenius-Perron operator as described in Section 2.3. Figures 9(b) and 9(a) show the Ulam-Galerkin matrix generated by the exact velocity fields in Eq. 49 and by the approximated optical flow field, respectively. As seen in Figure 3(a) there exist four basins and hence we may expect the Ulam-Galerkin matrix to be diagonalizable to four diagonal blocks. To reveal this



(a) A plot of the velocity field $u = -sin(2\pi x)cos(2\pi y)$ and its approximation from the image sequence

(b) A plot of velocity field $v = cos(2\pi x)sin(2\pi y)$ and its approximation from the image sequence

Figure 7: Comparison between the exact and approximated velocity fields from the image sequence using the smoothness parameter $\alpha = \beta = 0.01$. The transparent solid lines illustrate the exact velocity field and the colored plots represent the approximation.

correct block-diagonal form based on the modularity method we first partition the phase space into small triangles and index them to generate a graph network that represents the transport between these triangles. Then we apply the modularity method to discover the community structure of the graph network and use this structure to reorder the Ulam-Galerkin matrix. The results of the Ulam-Galerkin after reordering are shown in Figures 10(a) and 10(b). They reveal to us the four basins and the sorted Ulam-Galerkin matrix generated by the exact velocity is now in the correct block-diagonal form, whereas the sorted matrix generated by the optical flow is in the "almost" block-diagonal form due to the imperfection of the approximated velocity field that causes the transport between the basins.

5.2 Example 2

Our next example is a simulation result obtained from a numerical solution of the complex Ginzburg-Landau eqaution(CGLE). Figures 11 illustrates the initial distribution and a time series of brightness patterns. By a careful observation one may expect that there exists two sinks in the phase space located by the arrows in Figure 11(a). Since the analytic solution for this example is not priorly known, we ascertain our result only qualitatively.

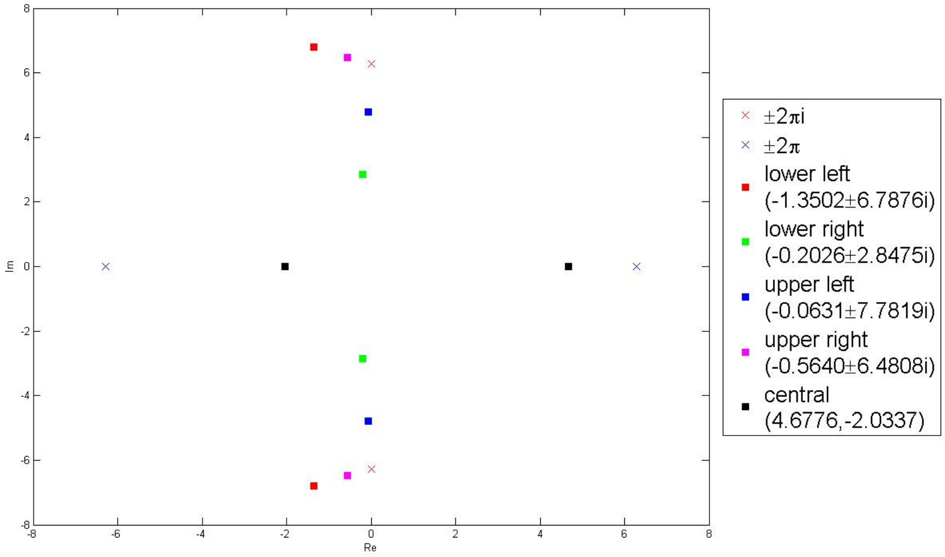
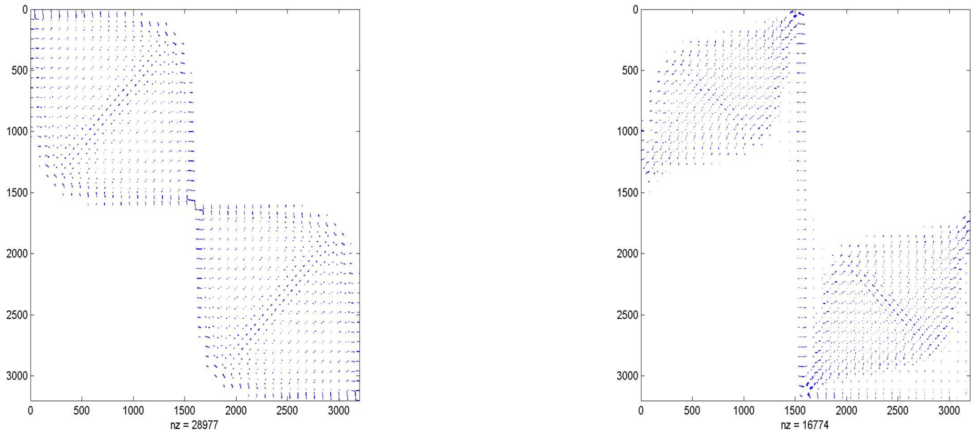


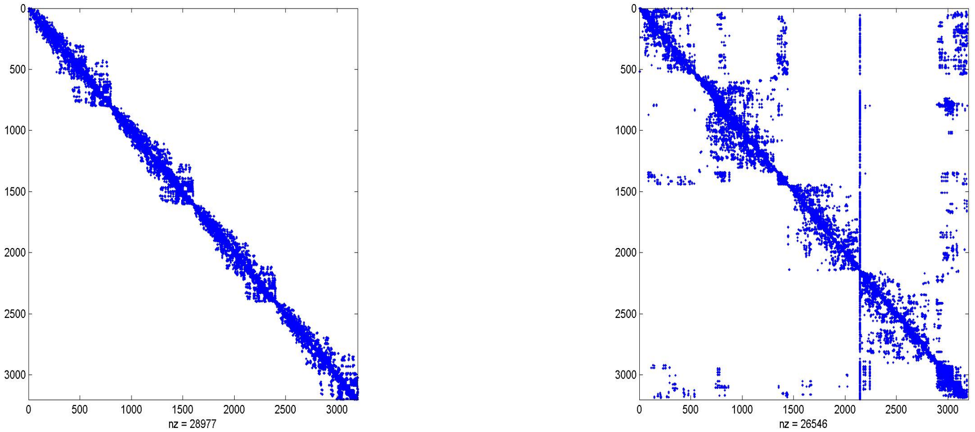
Figure 8: The plot shows the eigenvalues of the centers and the saddle node in the phase space of the exact velocity field in cross markers. The square markers locate the eigenvalues of those points computed from the estimated velocity field with the smoothness constraint $\alpha = 0.01$.



(a) The Ulam-Galerkin matrix generated by the exact velocity

(b) The Ulam-Galerkin matrix generated by the optical flow

Figure 9: The Ulam-Galerkin matrix generated by the exact velocity fields expressed in Equation 49 and by the approximated optical flow field using 200000 randomly chosen points.



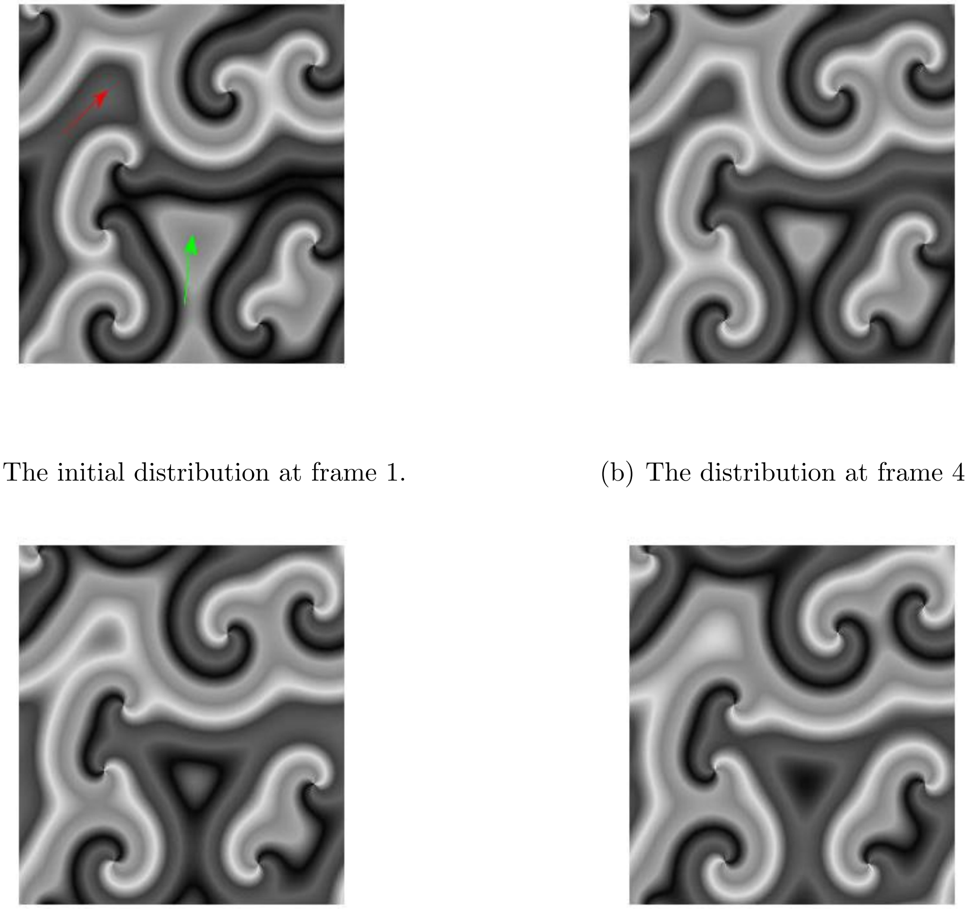
(a) The sorted Ulam-Galerkin matrix generated by the exact velocity

(b) The sorted Ulam-Galerkin matrix generated by the optical flow.

Figure 10: The Ulam-Galerkin matrix after sorting using the modularity method.

The estimated velocity fields with different values of parameters β but fixed $\alpha = 1$ are shown in Figure 12. We observe that when β is small, the velocity field at the region where occlusion occurs is strongly emphasized compared to other regions in the image whereas the magnitude of the velocity field is distributed more isotropically throughout the image region while increasing the value of β. To have a better view of this phenomenon, we plot in Figure 13 the contour line of the magnitude of the velocity onto the image of the initial distribution. We also notice that the velocity field becomes discontinuous where the occlusions take place. Nonetheless, we still observe the discontinuity of velocity field in those regions in the case of large β. Note that similar results can also be obtained when β is fixed and α is varied.

Next, we compute the matrix approximation of the Frobenius-Perron operator shown in Figure 14(a). Then we apply the modularity method to the graph generated from the semigroup of transformation of the ode (32) based on the estimated velocity field. We obtain two large communities and several trivially small communities from the result. Thus we group all those small communities together. Subsequently, we use the new ordering based on the community structure to reordered the Ulam-Galerkin matrix to reveal the “almost” block diagonal form, see Figure 14(b), and we also paint the phase space according the community structure discovered by the modularity method as shown in Figure 15, where the two large communities are painted in different colors and those small communities are not painted. Notice that there are three blocks in sorted matrix, where the middle block corresponds to those



(a) The initial distribution at frame 1.

(b) The distribution at frame 40.

(c) The distribution at frame 80.

(d) The distribution at frame 120.

Figure 11: A sequence of some images captured from the numerical simulation of the Complex Ginzburg-Landau equation. The arrows point at the sinks that can be found through a careful observation and as revealed by the Ulam-Galerkin matrix analysis as in Figure 12- 15

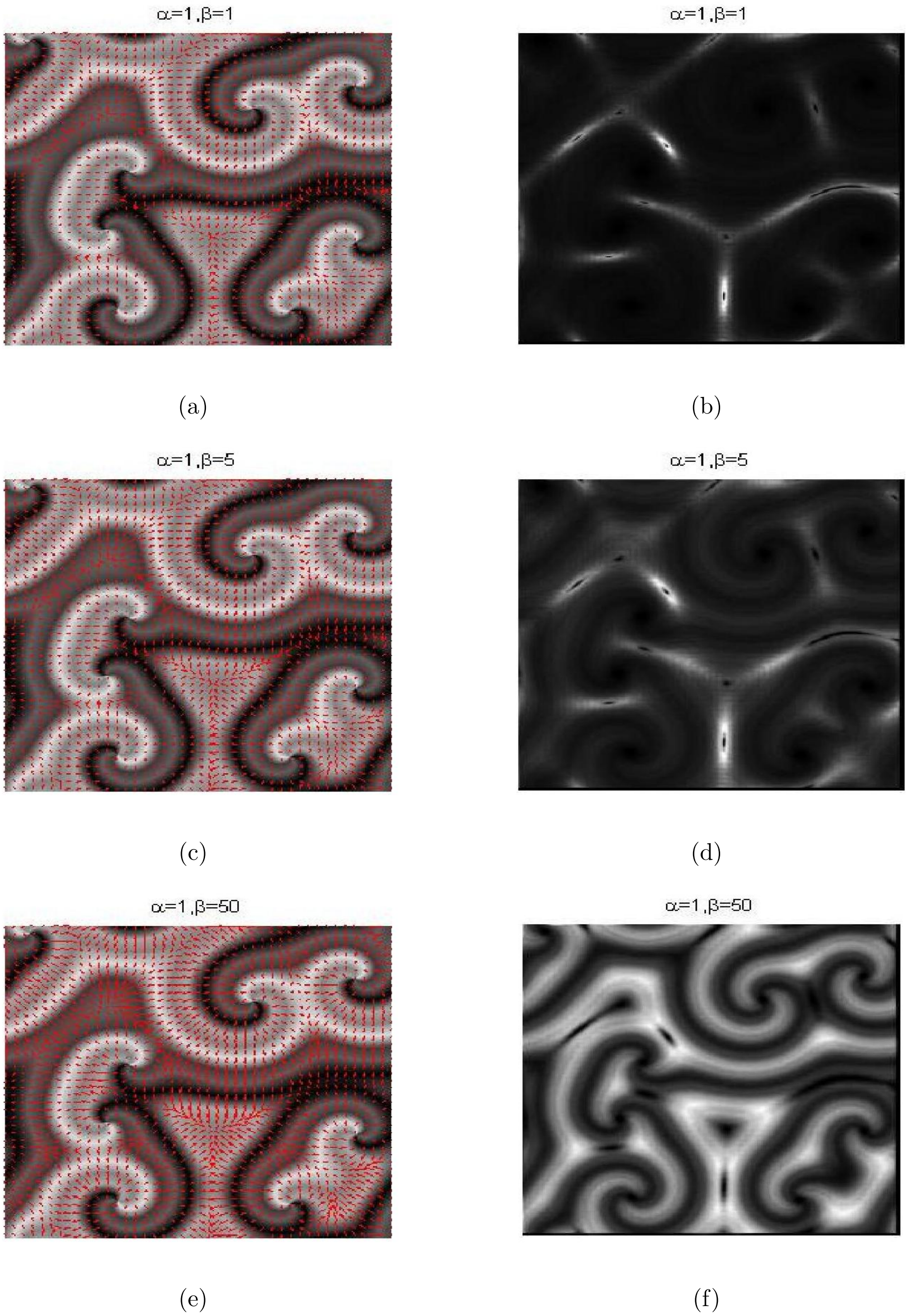
of several small communities that we group them together and the other two blocks represent the two large communities.

6 Conclusion

In this paper, we have presented a computational method to construct the finite-rank approximation of the Frobenius-Perron operator from successive images, which essentially requires a technique to recover velocity fields from image sequences. We have related this to the famous optical flow problem in various forms. Thus we present a mathematical model to this problem and we demonstrate its validity based on the theory of the infinitesimal generator of the Frobenius-Perron operator. We employ a regularization method to solve an ill-posed problem to reconstruct the desired velocity field. Subsequently, we resort to the resultant velocity field to identify number and location of the almost invariant sets using the graph modularity method. We have demonstrated our methods on synthetic data to demonstrate their usefulness to study a qualitative behavior, transport phenomena, and other ergodic properties of measurable dynamical systems captured through image sequences.

7 Acknowledgements

We thank Aaron Clauset and Mark Newman for sharing their codes used in [16, 17] for the modularity method computations. We are grateful to the National Science Foundation of the USA for their support of both of the authors of this pro ject through DMS-0404778.

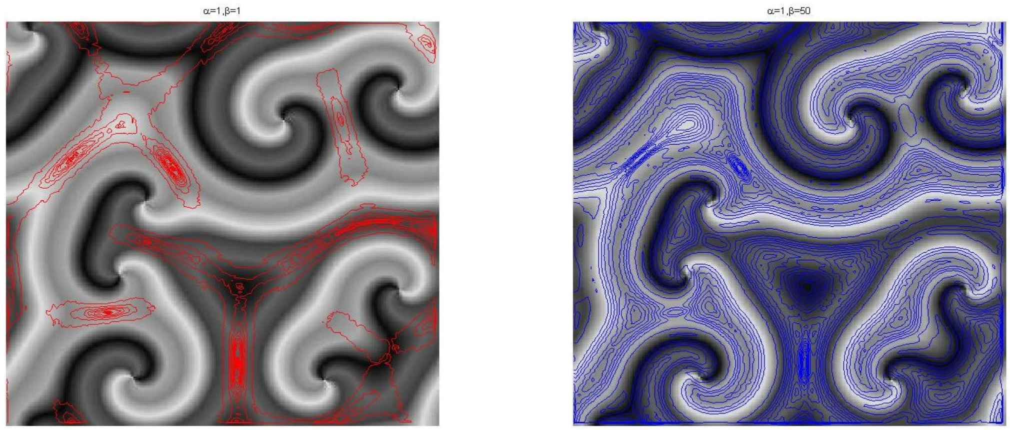


(a) (b)

(c) (d)

(e) (f)

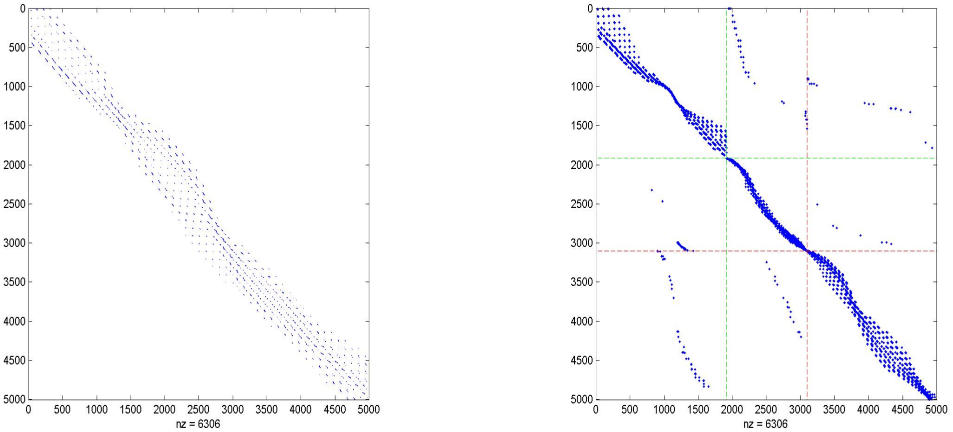
Figure 12: The velocity fields approximated from the div-curl regularization with various values of parameters α and β are plotted side by side with the corresponding magnitude of the velocity fields.



(a)

(b)

Figure 13: (a) For a small β at $\alpha = 1$, the magnitude of the velocity field is more emphasized at the region where occlusion occurs. In this region the velocity become discontinuous. (b) For a large β at $\alpha = 1$, we observe that the velocity field is isotropically distributed throughout the image region. Nonetheless, the discontinuity can still be observed at the region where occlusion take places.



(a) The unsorted Ulam-Galerkin matrix

(b) The sorted Ulam-Galerkin matrix

Figure 14: The Ulam-Galerkin matrix before and after sorting. Notice that the "almost" block diagonal form is unveiled after sorting. The middle block corresponds to those of several small communities that we group them together. The other blocks represent the two large communities.

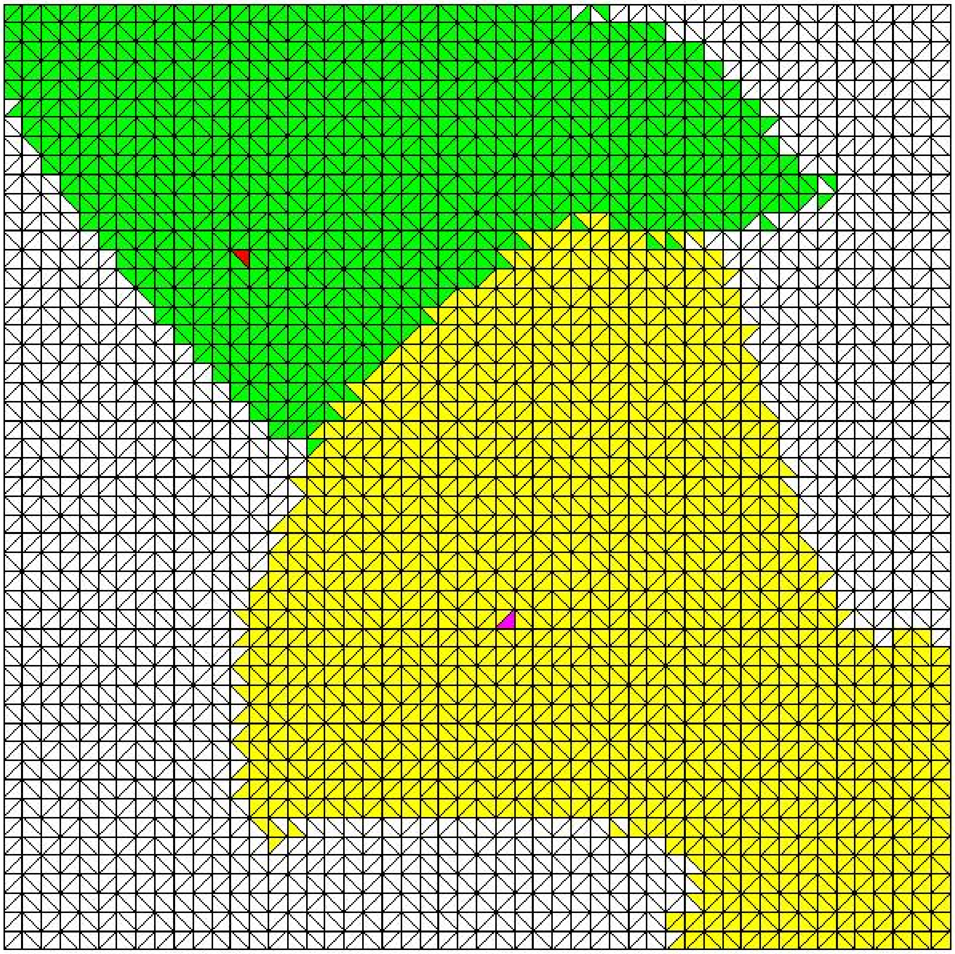


Figure 15: The phase space are partitioned into three regions corresponding two basins of the sinks corresponding to two communities discovered by the modularity method and the region that an initial point does not converge to neither basins.

References

[1] E. Bollt, L. Billings, and I. Schwartz. A manifold independent approach to understanding transport in stochastic dynamical systems. *Physica D*, 173:153–177, 2002.

[2] L. Billings, E. Bollt, and I.B. Schwartz. Phase-space transport of stochastic chaos in population dynamics of virus spread. *Physical Review Letters*, 88(234101), 2002.

[3] G. Froyland. Extracting dynamical behaviour via markov models. In *Proc. Nonlinear dynamics and statistics*, pages 283–324, Boston, MA, 2001. Birkhauser.

[4] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial intelligence*, 17:185–203, 1981.

[5] T.Corpetti, E.Měmin, and P.Pěrez. Dense estimation of fluid flows. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(3):365–380, 2002.

[6] T.Corpetti, D.Heitz, G. Arroyo, E.Memin, and A.Sant-Cruz. Fluid experimental flow estimation based on an optical-flow scheme. *Experiments in Fluids*, 40:80–97, 2006.

[7] D. Bereziat and J.-P. Berroir. Motion estimation on meteorological infrared data using a total brightness invariance hypothesis. *Env. Mod. Soft.*, 15.

[8] J.J. Simpson and J.I. Gobat. Robust velocity estimates, stream functions, and simulated lagrangian drifters from sequential spacecraft data. *IEEE Transactions on Geoscience and Remote Sensing*, 32(3).

[9] P. Heas, E. Memin, and N. Papadakis. Dense estimation of layer motions in the atmosphere. In *Int. Conf. Pattern Recognition (ICPR' 06)*, Hong-Kong, China, 2006.

[10] S. Gutierrez and D.G. Long. Optical flow and scale-space theory applied to sea-ice motion estimation in antarctica. *Geoscience and Remote Sensing Symposium, 2003. IGARSS '03. Proceedings.*, 4:2805– 2807, 2003.

[11] A. Suvichakorn and A. Tatnall. The application of cloud texture and motion derived from geostationary satellite images in rain estimation - a study on mid-latitude depressions. In *Geoscience and Remote Sensing Symposium, 2005. IGARSS apos;05. Proceedings*, volume 3.

[12] R.P.Wildes, M.J.Ambabile, A.M. Lanzillotto, and T.S. Leu. Recovering estimate of fluid flow from image sequence data. *Computer vision and Image understanding*, 80:246–266, 2000.

[13] A.Lasota and M.C.Mackey. *Chaos, fractals, and noise, Stochastic Aspects of Dynamics.* Springer, New York, 2 edition, 1994.

[14] M.E.J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69(026113), 2004.

[15] M.E.J. Newman and M. Girvan. *Statistical Mechanics of Complex Networks.* Springer, Berlin, 2004.

[16] M.E.J. Newman. Fast algorithm for detecting community structure in networks. *Phys. Rev. E*, 69(066133), 2004.

[17] Aaron Clauset, M. E. J. Newman, and Cristopher Moore. Finding community structure in very large networks. *Phys. Rev. E*, 70(066111), 2004.

[18] M. Dellnitz and O. Junge. On the approximation of complicated dynamical behavior. *SIAM Journal on Numerical Analysis*, 36(2):491–515, 1999.

[19] R. Preis and M. Dellnitz. *Congestion and Almost Invariant Sets in Dynamical Systems.* Proceedings of Symbolic and Numerical Scientific Computation (SNSC'01). Springer, 2003.

[20] R. Preis, M. Dellnitz, M. Hessel, Ch. Schtte, and E. Meerbach. Dominant paths between almost invariant sets of dynamical systems. *Preprint 154 of the DFG Schwerpunktprogramm 1095*, 2004.

[21] K. Padberg, R. Preis, and M. Dellnitz. Integrating multilevel graph partitioning with hierarchical set oriented methods for the analysis of dynamical systems. *Preprint 152 of the DFG Schwerpunktprogramm 1095*, 2004.

[22] S.M. Ulam. *Problems in Modern Mathematics.* Science Editions. Wiley, New York, 1970.

[23] T.Y. Li. Finite approximation for the Frobenius-Perron operator. a solution to Ulam's conjecture. *J. Approx. Theory*, 17(2):177–186, 1976.

[24] C. Chui, Q. Du, and T. Li. Error estimates of the Markov finite approximation of the Frobenius Perron operator. *Nonlinear Analysis*, 19(4):291–308, 1992.

[25] J. Ding and A. Zhou. The projection method for computing multidimensional absolutely continuous invariant measures. *J. of Statistical Physics*, 77(3/4):899–908, 1994.

[26] G. Froyland. Finite approximation of Sinai-Bowen-Ruelle measure for Anosov systems in two dimension. *Random Comput. Dynam.*, 3(4):251–263, 1995.

[27] F. Hunt. Approximating the invariant measures of randomly perturbed dissipative maps. *J. Math. Anal. Appl.*, 198(2):534–551, 1996.

[28] G. Froyland. *Estimating Physical Invariant Measures and Space Averages of Dynamical Systems Indicators.* PhD thesis, University of Western Australia, 1996.

[29] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI81*, pages 674–679, 1981.

[30] J. Bign, G.H. Granlund, and J. Wiklund. Multidimensional orientation estimation with applications to texture analysis and optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(8):775–790, 1991.

[31] J.L.Barron, D.J.Fleet, and D.J.Beauchemin. Performance of optical flow techniques. *Int. J. Comput. Vis.*, 12(1):43–77, 1994.

[32] B. Galvin, B. McCane, K. Novins, D. Mason, and S. Mills. Recovering motion fields: An evaluation of eight optical flow algorithms. In *In Proc. the Ninth British Machine Vision Conference (BMVC '98)*, volume 1, pages 195–204, 1998.

[33] A.N.Tikhonov and V.Y.Arsenin. *Solutions of ill-posed problem.* Winston, Washington, DC, 1977.

[34] D. Suter. Motion estimation and vector splines. pages 939–942, 1994.

[35] E.P.Simoncelli. Design of multi-dimensional derivative filters. *IEEE Int. Conf. Image Processing*, 1:790–793, 1994.

[36] M.W.Hirchh and S.Smale. *Differential Eqations, Dynamical Systems, and Linear Algebra.* Academic Press, Orlando, 1974.

[37] J.P. Bagrow and E. M. Bollt. A local method for detecting communities. *Phys. Rev. E*, 72(046108), 2005.

[38] Joseph D. Skufca and Erik M. Bollt. Mostly conjugate: Relating dynamical systems — beyond homeomorphism. *submitted to SIAM Journal on Applied Dynamical Systems*, 2006.