

Filtrado Adaptativo Dinámico para la Visualización de Campos Vectoriales

Julián Dominguez

Claudio Delrieux

Departamento de Ingeniería Eléctrica - Universidad Nacional del Sur
Av. Alem 1253, (8000) Bahía Blanca, Argentina
jdoming@uns.edu.ar, claudio@acm.org

Abstract:

La visualización científica encuentra una especial aplicación en el estudio de la dinámica de los sistemas no lineales. Esencialmente, los sistemas de dinámica compleja son de difícil o imposible resolución analítica, quedando como único camino su análisis por métodos gráficos para el entendimiento de su dinámica. Para los métodos de visualización se han creado diversas metáforas que permiten una rápida comprensión del sistema. Entre ellas ocupan un lugar destacado los Streamlines y la LIC, con diversas variantes y optimizaciones en tiempo de cálculo y calidad de resultado. En el presente trabajo presentaremos una nueva técnica de visualización de sistemas dinámicos basado en la utilización de un filtrado espacial direccional adaptativo. Este método puede ser utilizado por separado, pero sin embargo veremos que su mayor utilidad radica en el refuerzo de la eficacia obtenida junto con otros métodos basados en texturas como LIC o CLIC.

Palabras Clave: VISUALIZACIÓN CIENTÍFICA — SISTEMAS NO LINEALES Y CAÓTICOS — (LIC) LINE INTEGRAL CONVOLUTION — PROCESAMIENTO DE IMÁGENES.

1 Introducción

Durante la última década la visualización científica se ha tornado de particular importancia para el análisis de la mecánica del continuo, en particular la dinámica

de sistemas. Este auge se debe principalmente a la aparición de teorías modernas que muestran que es virtualmente imposible hallar una solución analítica general para un sistema dinámico de orden 3 o mayor, donde aparece la posibilidad de “Caos” en el mismo (siendo aún de difícil resolución para sistemas de segundo orden) [12]. En la mayoría de los casos, estos sistemas no son integrables en forma analítica, por lo que su adecuada comprensión solo puede realizarse por medio de simulaciones computacionales, las cuales se representan de un modo natural y eficiente por medios gráficos.

La necesidad de obtener una idea conceptual del comportamiento de un sistema dinámico, a partir de sus ecuaciones diferenciales, ha llevado a la creación de numerosas técnicas y metáforas con el fin de caracterizarla lo más fielmente posible a partir de cualquier condición inicial. Podemos mencionar la tradicional y ya desactualizada idea de los vectores como *flechas*, la cual codifica los vectores tangentes al campo en sentido y magnitud, hasta llegar a ideas más modernas, como la *convolución lineal* de una textura en función del campo vectorial asociado (LIC), pasando por métodos intermedios como Streamlines, Spot Noise, y otras variantes de tales métodos, los cuales ofrecen una riqueza mucho mayor, así como mejor calidad de la visualización (ver por ejemplo [1, 2, 3, 9, 13, 14]). Las técnicas desarrolladas para estos objetivos son de gran importancia teórica y experimental en diversas disciplinas, como por ejemplo dinámica de los fluidos, ecología, electrónica, mecánica no lineal, etc. Estas aplicaciones han ido cobrando importancia en la misma medida en que los sistemas dinámicos han sido incorporados y utilizados como modelo matemático mediante sus ecuaciones diferenciales.

En este trabajo presentaremos un nuevo acercamiento, basado en la utilización de filtrado direccional adaptativo. Este método puede ser utilizado por separado, pero ofrece una mayor utilidad y eficacia en forma conjunta con otros métodos basados en texturas como LIC o CLIC. En la siguiente Sección presentamos una reseña de los métodos basados en texturas (los de mayor importancia para nuestro desarrollo). En la Sección 3 presentamos la idea de los filtros direccionales adaptativos, y su utilización como método de visualización. También mostramos de qué manera esta técnica puede ser utilizada en forma conjunta con los métodos basados en texturas. En la Sección 4, por último, presentamos las conclusiones y los trabajos futuros.

2 Visualización basada en texturas

La idea más básica consiste en visualizar el campo vectorial originado por el sistema dinámico por medio de objetos de orden 0 (vectores). De esa manera, el dominio matemático de interés (espacio de fases), es recorrido a intervalos regulares, evaluándose el sistema dinámico en cada uno de esos puntos, y graficando en ellos un vector con magnitud y dirección tomadas de dicha evaluación. Este método es rápido y sencillo, pero de muy escasa utilidad. Una idea un poco más elaborada

consiste en utilizar objetos integrales (*streamlines*), los cuales se pueden obtener por medio de una integración numérica del sistema. Al igual que en el caso anterior, se recorre el espacio de fases en forma regular, donde ahora cada punto de la recorrida es una condición inicial o *semilla*, a partir de la cual el sistema va a ser integrado numéricamente. Esta integración se realiza una cantidad arbitraria de pasos, y la trayectoria resultante es graficada. El conjunto de trayectorias así obtenidas puede dar una muy buena idea de todo el conjunto posible de trayectorias en el sistema (o sea, su “diagrama de fases”).

Los diagramas obtenidos por medio de trayectorias o streamlines tienen las ventajas de ser también muy sencillos de implementar y rápidos en su ejecución. Sin embargo, requieren del usuario la elección de varios parámetros para producir resultados útiles, básicamente la densidad del conjunto de semillas, la cantidad de pasos de integración por trayectoria, y el diferencial finito utilizado. Una elección inadecuada de estos parámetros hace que los resultados visuales obtenidos sean muy pobres. Por lo tanto se requiere cierto grado de experiencia en el usuario para que el método sea adecuado. Además, ciertos problemas como por ejemplo la imposibilidad de representar puntos críticos inestables o de estabilidad mixta, la desapareja densidad de las visualizaciones del diagrama de fases, y otros, hace que la aplicabilidad de los streamlines como método científico sea muy limitada.

A partir de trabajos originados hace una década, el área de visualización de sistemas dinámicos experimentó un notable influjo de propuestas de métodos y técnicas. Todas estas se basan en la LIC o *Line Integral Convolution* [1], la cual consiste en convolucionar una textura de entrada a través de la trayectoria recorrida desde una condición inicial y a lo largo de la trayectoria integral originada en dicha semilla. En la gran mayoría de los trabajos, se utiliza como textura de entrada un *ruido blanco* (la intensidad, crominancia y saturación de cada *texel* o pixel de la textura son aleatorios y uniformemente distribuidos). El resultado de integrar esta textura a lo largo de la trayectoria originada en un punto del espacio de fases produce como resultado un color que está espacial y direccionalmente correlacionado con la dirección y velocidad que el espacio vectorial tiene en ese punto. Por lo tanto, si bien el cómputo a realizar es bastante mayor, todas las dificultades y desventajas de los streamlines se solucionan. La visualización que se obtiene con la LIC es casi siempre adecuada, no tiene zonas sin cubrir, es bastante robusta con respecto a los parámetros de la evaluación numérica, etc. Su única desventaja es el tiempo de cómputo, que es de varios órdenes de magnitud mayor. Por dicha razón, en esta década surgieron muchos métodos que buscan reducir el tiempo de cómputo de la LIC, así como enriquecer la calidad de las visualizaciones tratando de ilustrar otras características temporales o diferenciales del sistema.

En esa línea, y basándonos en la semejanza entre *Streamlines* y *LIC* presentamos en trabajos anteriores un conjunto de técnicas que reúnen las ventajas de ambas, en particular rapidez y calidad, y fundamentalmente que permiten negociar la precisión en función del tiempo de cómputo. Uno de estos métodos es el *CLIC* o *Cumulative*

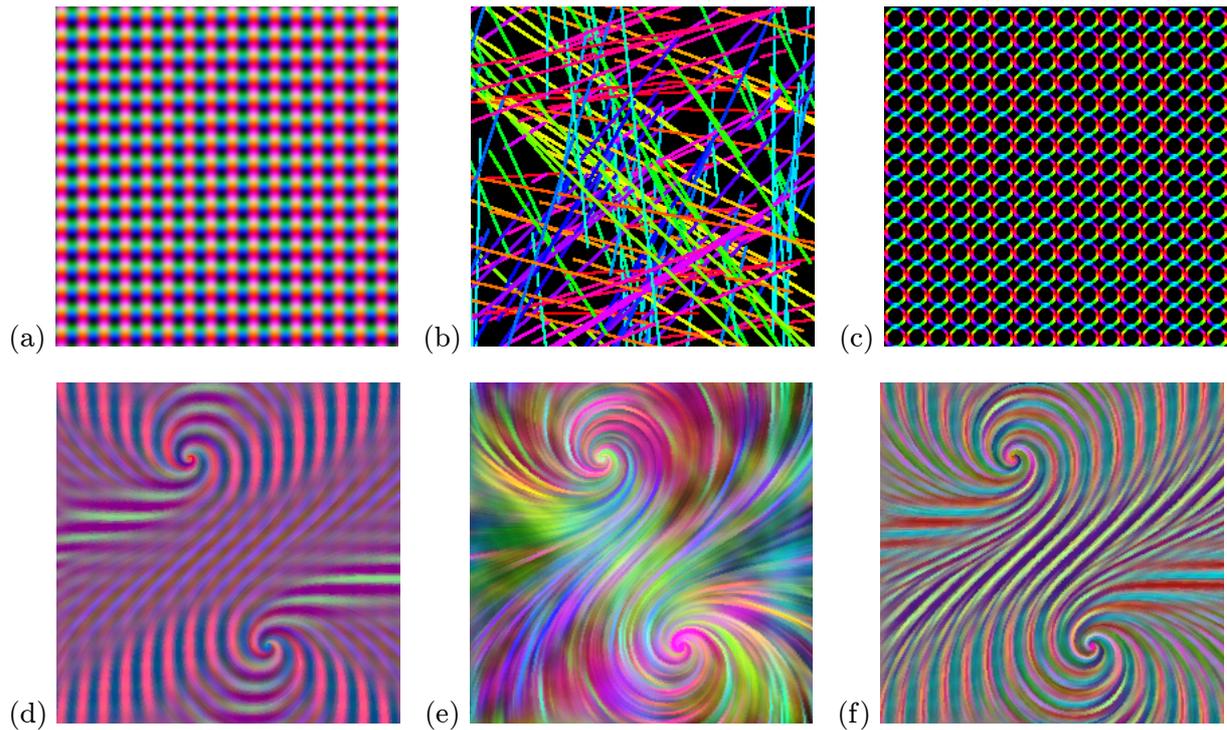


Figura 1: Tres texturas diferentes, donde todas tienen codificado en sus colores la dirección, y los resultados de convolucionarlas en el mismo sistema utilizando el algoritmo Doble CLIC.

Line Integral Convolution [4, 5]. A su vez, hemos destacado la importancia que tiene la composición de la textura para resaltar las características del sistema en estos métodos basados en texturas. En efecto, la textura constituye un medio para introducir determinadas características geométricas y cromáticas en la imagen final, las cuales interactúan con la distribución geométrica de las trayectorias en el diagrama de fases. La LIC, así como el CLIC, interactúan con la textura, realizando un “esfumado” direccional en el sentido del campo vectorial (ver Fig. 1).

La importancia y utilidad de este aspecto para la eficacia de la visualización ha sido poco estudiado (con excepción de [10] donde se considera solamente la distribución espectral de diversos ruidos utilizados como textura). El ruido blanco de alta frecuencia como textura de entrada permite resaltar los detalles finos en el campo vectorial, mientras que el mismo ruido filtrado con un pasabajos tiende a destacar la forma global del diagrama de fases. En [4, 5] se han sugerido algunas técnicas para generar texturas más adecuadas, las cuales luego pueden ser utilizadas para resaltar diversas características de un determinado campo vectorial.

Otra información útil en el análisis de los sistemas no lineales es la relativa a dirección y magnitud del campo vectorial en cada punto del sistema. Para ello, desarrollamos una representación que permite visualizar ambas magnitudes simultáneamente, denominada “*Color Map*”, la cual asocia la dirección del campo vectorial con la luminancia en el diagrama de fase, y la crominancia con la velocidad escalar. En

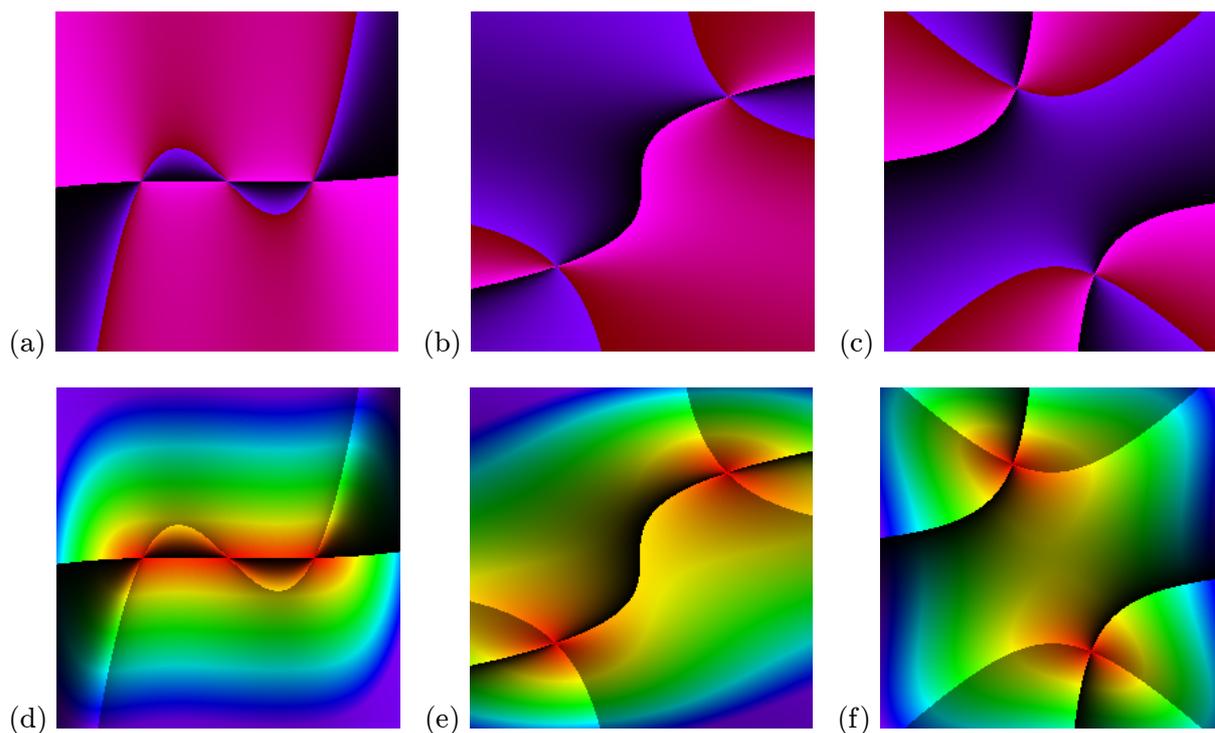


Figura 2: Tres sistemas descritos por su *Color Map* asociado. Se puede apreciar los puntos críticos en el cruce de las isóclinas.

la Fig. 2 podemos observar los resultados obtenidos con esta técnica en algunos sistemas dinámicos. Este mapeo permite reconocer las isóclinas (líneas de igual dirección en el sistema) principales: la de 0° y la de 90° , las cuales proveen información de sumo interés. El cruce de estas isóclinas determina la ubicación de los puntos críticos.

3 Filtrado espacial adaptativo

Como pudimos ver en la Sección anterior, la LIC resalta las características de la textura cuya dirección coincide con la dirección del campo vectorial en un punto dado. Sin embargo, y como veremos a continuación, ésta no es la única manera de producir tal resultado. El filtrado espacial por convolución de una imagen se basa en la idea de una integración pesada por un *kernel de convolución* [8, 7], de similar manera a la vista en la LIC con su integral lineal. Los filtros de convolución son ampliamente utilizados en el área del procesamiento de imágenes para el filtrado de ruido, en compresión de imágenes, video, comunicación, etc.

El filtrado por convolución es capaz de producir excelentes resultados aún con kernels chicos (3×3), lo que produce el efecto esperado a un bajo costo computacional. El filtrado para resaltar bordes es un ejemplo de filtrado direccional. A continuación,

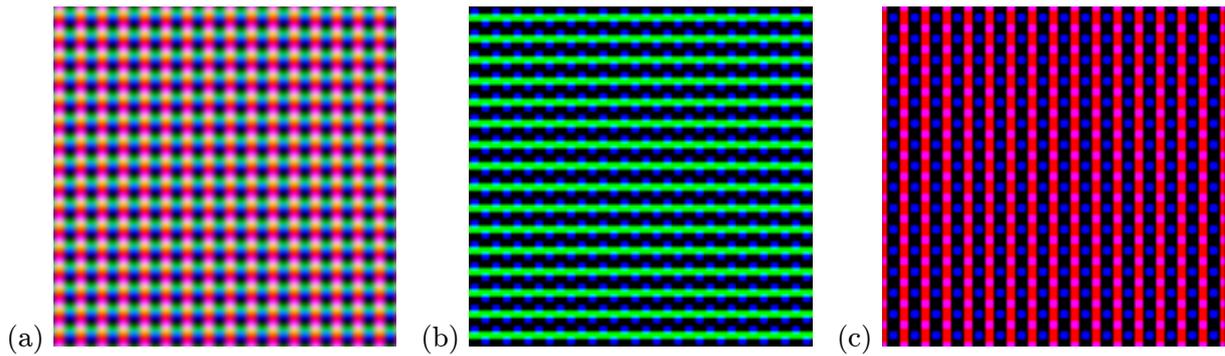


Figura 3: A partir de *a*) obtenemos la texturas filtradas horizontal (figura *b*, campo a 0°) y verticalmente (figura *c*, campo a 90°).

mostramos cómo un kernel estático nos permite visualizar una dirección determinada que se coincide en cada punto con la dirección en la textura de entrada. En este caso es un filtrado horizontal (o vertical si rotamos 90° la matriz o *kernel*), denominado filtro de Sobel [11, 6].

$$\begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix} = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$$

En la Fig. 3 podemos observar una textura filtrada horizontal y verticalmente con este filtro. Como vemos, en esta textura son especialmente visibles las direcciones principales, pues fue diseñada para resaltarlas (en rojo y verde), además de ambas diagonales (en azul). Nuestro interés radica en resaltar la dirección de cada punto dentro del diagrama de fase. De esa manera surge naturalmente la idea de aplicar un filtrado adaptativo de acuerdo a la dinámica local del sistema. Así como la LIC produce una convolución de línea sobre las trayectorias, el *Filtrado Adaptativo Dinámico* (o *FAD*) produce una convolución espacial en el entorno del punto analizado (ya no sólo en la dirección del campo vectorial), obteniendo un resultado suave (dependiendo del filtro), y rico en información de la dinámica local.

En otras palabras, el método se basa en tomar como patrón un kernel de filtrado horizontal (0°) y luego exponerlo a la rotación local en cada punto, y filtrar en ese lugar la textura para resaltar la dinámica local. La implementación adoptada produce un salto entre $-\pi/2$ y $\pi/2$, implicando una una rotación de 180° en el filtro de convolución al pasar por los 90° , determinando que se visualicen las isóclinas de 90° , ayudando en la interpretación del resultado. En la Fig. 4 podemos ver algunos ejemplos, donde la textura de entrada (Fig. 3(a)) es convolucionada respectivamente con los siguientes filtros:

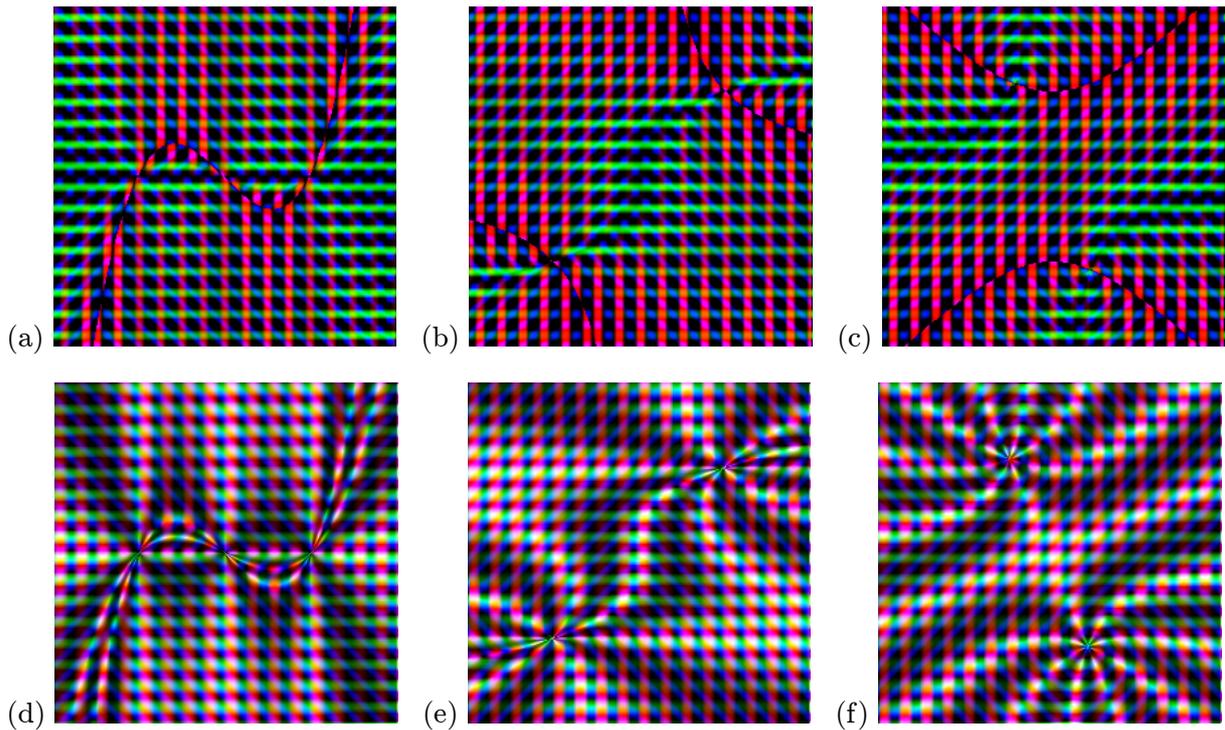


Figura 4: Dos secuencias de los mismos sistemas convolucionados con el *Filtro 1* (a, b y c), y el *Filtro 2* (d, e y f). El primero brinda mayor información respecto a los gradientes de dirección y trayectorias, mientras que el segundo resalta más la información de las isóclinas y puntos críticos.

$$\text{Filtro 1 : } \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}, \quad \text{Filtro 2: } \begin{pmatrix} -1 & -1 & -1 \\ 2 & 3 & 2 \\ -1 & -1 & -1 \end{pmatrix}$$

Se puede observar que el segundo kernel no produce discontinuidades en la visualización, debido a su simetría respecto a la horizontal. El resultado aún carece de las virtudes necesarias como para que el usuario obtenga una buena comprensión global del sistema al visualizarlo. Sin embargo, la combinación entre este filtrado direccional adaptativo y los métodos de texturas ya vistos es bastante sencilla e inmediata. Como ya se mostrara en trabajos anteriores, la *iteración* de un método (el CLIC en nuestro caso) es una forma rápida y directa de mejorar los resultados. La iteración consiste en utilizar la representación obtenida para el sistema como textura de entrada. La iteración en la LIC no produce mejoras en el resultado final. En la CLIC se producen mejoras notables con una sola iteración, pero no más allá.

La idea propuesta para explotar las ventajas combinadas de los filtros direccionales adaptativos y la CLIC consiste en aplicar alternativamente cada método. Se genera una imagen a partir de aplicar uno de los métodos a la textura de entrada, y tal resultado es utilizado como textura de entrada para el otro. Es de destacar que

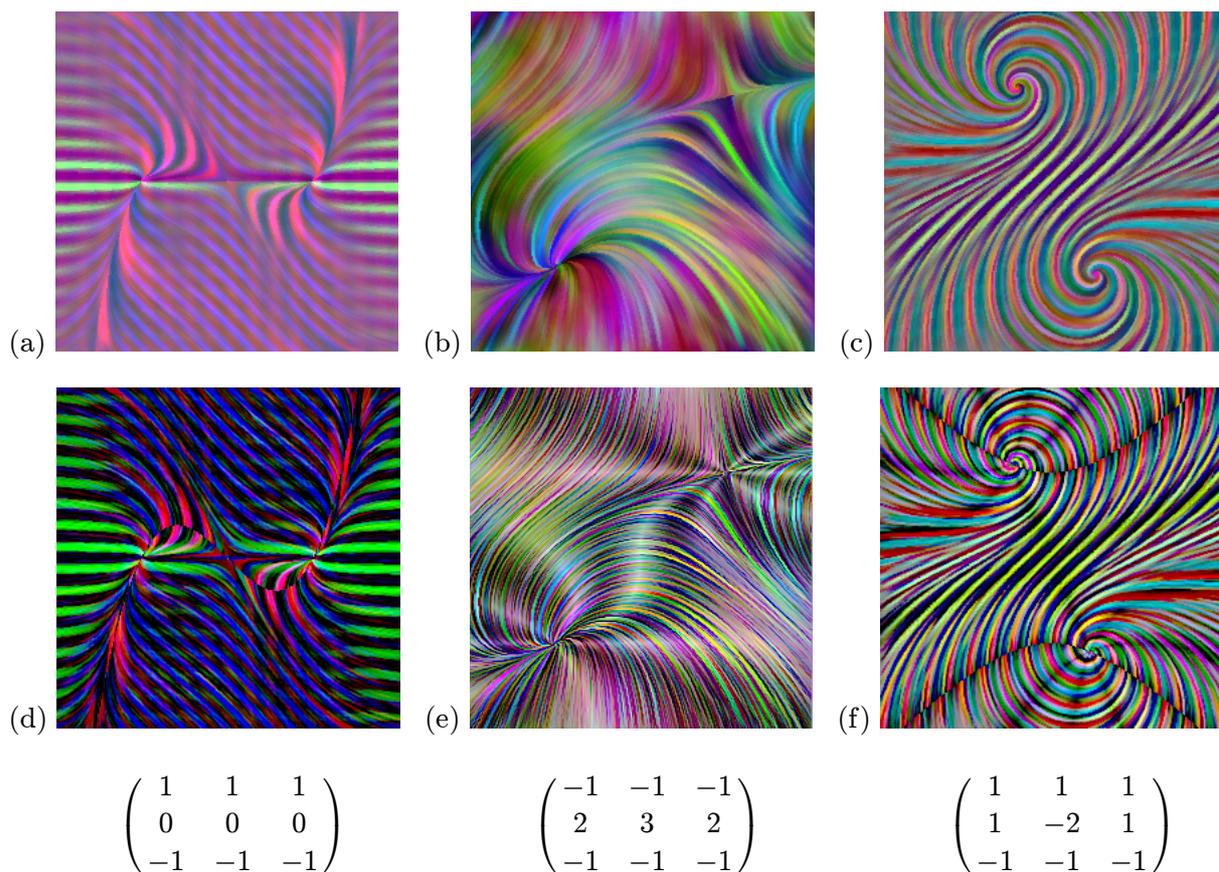


Figura 5: Los tres sistemas utilizando sólo Doble CLIC (en la primera fila), y aplicando CF con los kernels indicados.

cada una de ellas iterada independientemente de la otra no produce mejoras en la visualización, pero sí lo hace su alternancia, debido a que ambos métodos destacan propiedades diferentes en la visualización del sistema dinámico. Las posibilidades de combinación son innumerables, pero en este trabajo nos concentraremos en las secuencias de dos pasos filtrado + CLIC, CLIC + filtrado (denominadas FC y CF), y en las secuencias de tres pasos FCF y CFC.

Las evaluaciones de dos pasos son más rápidas (toda la operación toma entre 1 y 2 segundos utilizando Doble CLIC, ya que el filtrado por convolución es muy veloz). Cada operación tipo C (convolución lineal) suaviza el resultado, dándole a la visualización continuidad y coherencia. En cambio, las operaciones tipo F resaltan el comportamiento del sistema dinámico, provocando posibles discontinuidades en el sentido perpendicular al campo vectorial. Por ello es que se debe ser cuidadoso al momento de seleccionar el kernel del filtro para el último paso F, si es que hemos elegido un gráfico tipo CF o FCF (ver Fig. 5). En las Fig. 6 podemos observar los resultados obtenidos con tres pasos, FCF y CFC. En la Fig. 7, por último, se muestran el efecto de utilizar filtros simétricos como el de Sobel en el paso F.

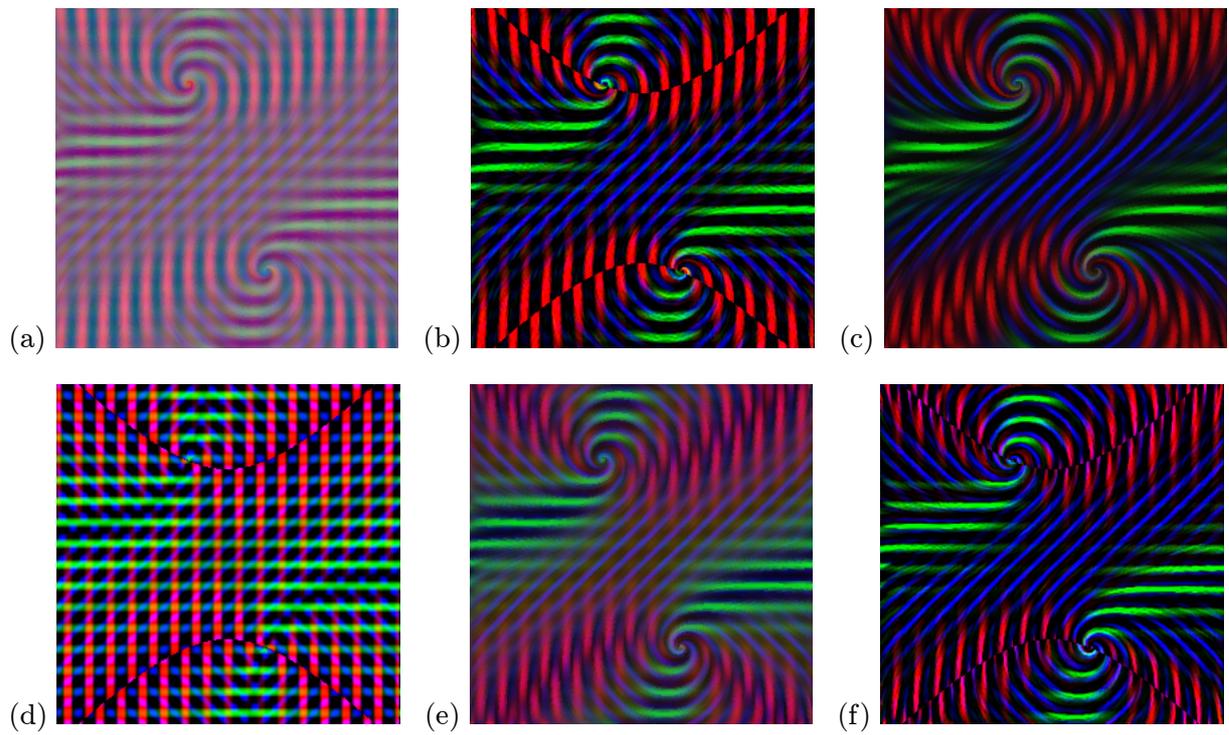


Figura 6: CFC en la primer fila y FCF en la segunda, utilizando filtro de Sobel.

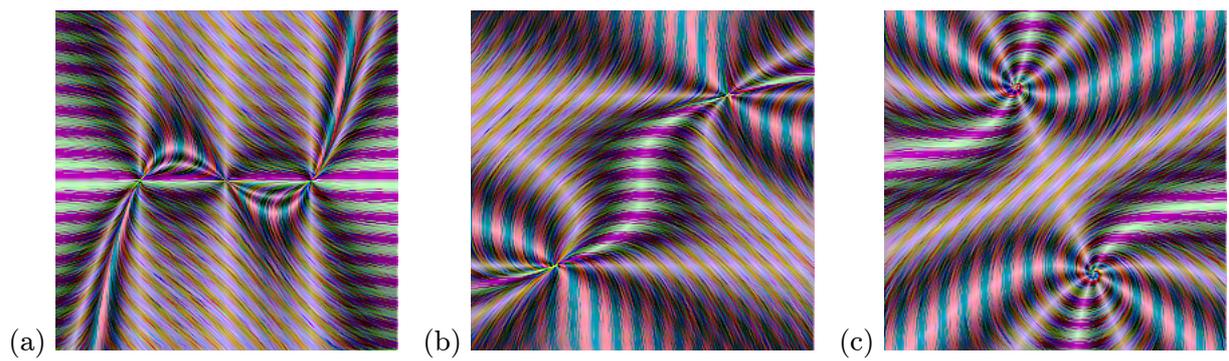


Figura 7: CF, utilizando para F el *Filtro 2* ya mencionado. Por ser un filtro simétrico suaviza aún más la etapa C, dando sensación de tridimensionalidad sin dejar de sugerir las isóclinas y puntos críticos.

4 Conclusiones y Trabajo Futuro

Presentamos un nuevo método para la visualización de sistemas dinámicos. El mismo que puede ser utilizado por separado, siendo sin embargo de mayor utilidad su uso combinado con los métodos basados en texturas como LIC o CLIC. La buena complementación con este tipo de métodos se debe a que utiliza información no solo en la dirección del campo vectorial (como los métodos de convolución lineal) sino también del entorno local. Esta técnica es de bajo costo, siendo rápida y mejorando notablemente la calidad de las imágenes obtenidas por los métodos basados en la LIC.

Es evidente que se pueden implementar un sinnúmero de variantes a partir de las ideas presentadas en la Sección 3, combinando diferentes kernels de filtrado con diferentes secuencias de evaluación. Otras ideas que pueden mejorar mucho la calidad de la visualización y que están siendo estudiadas son las siguientes:

- Utilizar un kernel de filtrado circular, que se deforme elípticamente en función de la velocidad local del sistema dinámico.
- Definir patrones adaptativos de filtrado que reflejen comportamientos cualitativos (nodo, silla, isóclinas principales) y lo destaquen por medio de mapeos visuales.
- Utilizar el mapa de direcciones e intensidades (obtenidos por ejemplo con el “Color Map”) y utilizar un kernel, rotándolo según la dirección local del sistema para codificar la dirección como un filtrado direccional. La velocidad local puede representarse como elipticidad (geoméricamente) o como intensidad (cromática).

Referencias

- [1] B. Cabral and L. Leedom. Imaging Vector Fields Using Line Integral Convolution. *ACM Computer Graphics (SIGGRAPH Proceedings)*, 25(3):263–270, 1993.
- [2] Roger Crawfis, Nelson Max, and Barry Becker. Vector Field Visualization. *IEEE Computer Graphics and Applications*, 14(5):50–56, 1994.
- [3] Thierry Delmarcelle and Lambertus Hesselink. Visualizing Second-Order Tensor Fields with Hypersrtreamlines. *IEEE Computer Graphics and Applications*, 13(4):25–33, 1993.
- [4] Claudio Delrieux, Julián Dominguez, and Andrés Repetto. Towards a CLIC in Vector Field Visualization. In *Proceedings of the CISST 2001 Conference*, pages 695–702, CSREA Press, ISBN 1-892512-73-4, 2001.

- [5] Claudio Delrieux, Juli n Dominguez, and Andr s Repetto. Advanced Techniques for Real-time Flow Visualization. In *SPIE Proceedings Vol. 4716*, pages 375–385, The International Society for Optical Engineering Press, ISBN: 0-8194-4466-9, www.spie.org/web/abstracts/4700/4716.html, 2002.
- [6] Andrew Glassner. *Principles of Digital Image Synthesis*. Morgan Kaufman, San Francisco, 1995.
- [7] Jonas Gomes and Luiz Velho. *Image Processing for Computer Graphics*. Springer, New York, 1997.
- [8] Rafael Gonzalez and Richard Woods. *Digital Image Processing*. Addison-Wesley, Wilmington, USA, 1996.
- [9] James Helman and Lambertus Hesselink. Visualizing Vector Field Topology in Fluid Flows. *IEEE Computer Graphics and Applications*, 11(3):36–46, 1991.
- [10] M.H. Kiu and D.C. Banks. Multi-Frequency Noise for LIC. In *Visualization '96 Conference Proceedings*, pages 121–126, Los Altos, CA, 1996. IEEE Press.
- [11] J. C. Russ. *The Image Processing Handbook*. CRC Press, Boca Raton, FL, 1989.
- [12] Stefen H. Strogatz. *Nonlinear Dynamics and Chaos*. Addison-Wesley, Reading, Massachusetts, 1994.
- [13] J. J. van Wijk. Spot Noise: Texture Synthesis for Data Visualization. *ACM Computer Graphics*, 25(4):309–318, 1991.
- [14] J. J. van Wijk. Flow Visualization with Surface Particles. *IEEE Computer Graphics & Applications*, 13(7):18–24, 1993.