

Nuevos Resultados en la Visualización de Campos Vectoriales

Julián Dominguez Claudio Delrieux Andrés Repetto

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

1 Visualización para el Estudio de la Mecánica del Continuo

La visualización científica es actualmente una de las alternativas de mayor importancia para el estudio de la mecánica del continuo, en particular la dinámica de sistemas. Este auge se debe principalmente a que es virtualmente imposible hallar soluciones analíticas generales para sistemas de relativamente baja complejidad [12]. En la mayoría de los casos, estos sistemas no son integrables en forma analítica, por lo que su estudio 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, pasando por métodos intermedios como Streamlines, Spot Noise, y otras variantes [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. En este trabajo presentamos los resultados más recientes de nuestro grupo de investigación, basados en la convolución de texturas y la utilización de filtrado direccional adaptativo. Si bien estos métodos pueden ser utilizado por separado, aquí mostramos que ofrecen una mayor utilidad y eficacia al ser aplicados en forma conjunta.

2 Convolución de Texturas

La visualización por medio objetos de orden 0 (vectores) u orden 1 (trayectorias o *streamlines*) tiene la ventaja de ser sencilla de implementar y rápida en su ejecución. Sin embargo, requieren mucha supervisión por parte del usuario (la elección de los parámetros de evaluación numérica, 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* sea muy limitada. La LIC o *Line Integral Convolution* [1]

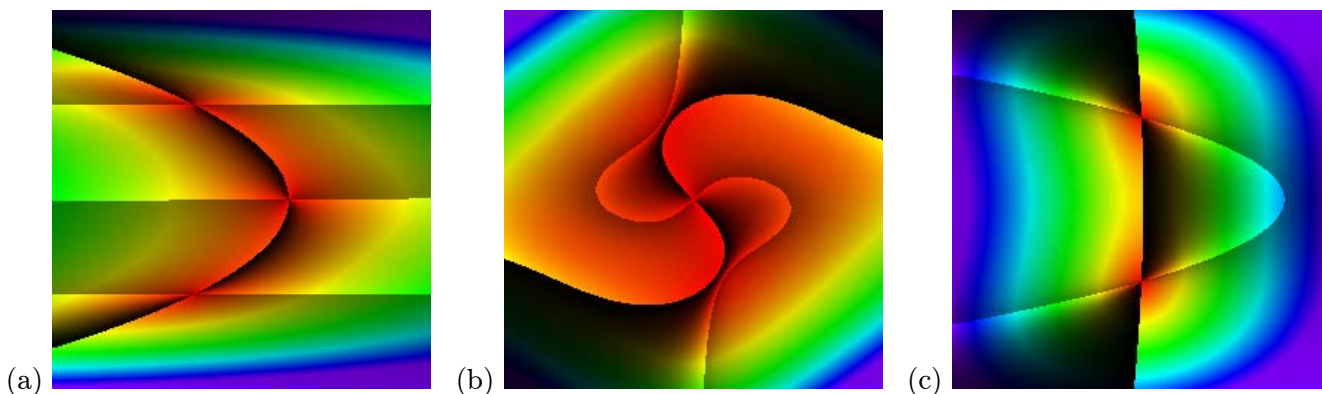


Figura 1: Tres sistemas visualizados con la paleta *Color Map*. Los puntos críticos se localizan en el cruce de las isóclinas.

representó un cambio notable en la calidad de estos sistemas. La LIC 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 sin supervisión, 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 este trabajo se proponen 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.

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 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. 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). 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 paleta bivariada que permite visualizar ambas magnitudes simultáneamente, denominada “*Color Map*”, la cual asocia la luminancia con la dirección del campo vectorial en el diagrama de fase, y la crominancia con la velocidad escalar. En la Fig. 1 podemos observar los resultados obtenidos con esta técnica en algunos sistemas dinámicos. Este mapeo permite reconocer las isóclinas principales (líneas de igual en las cuales una de las derivadas direccionales se anula), las cuales proveen información de sumo interés. El cruce de estas isóclinas determina la ubicación de los puntos críticos.

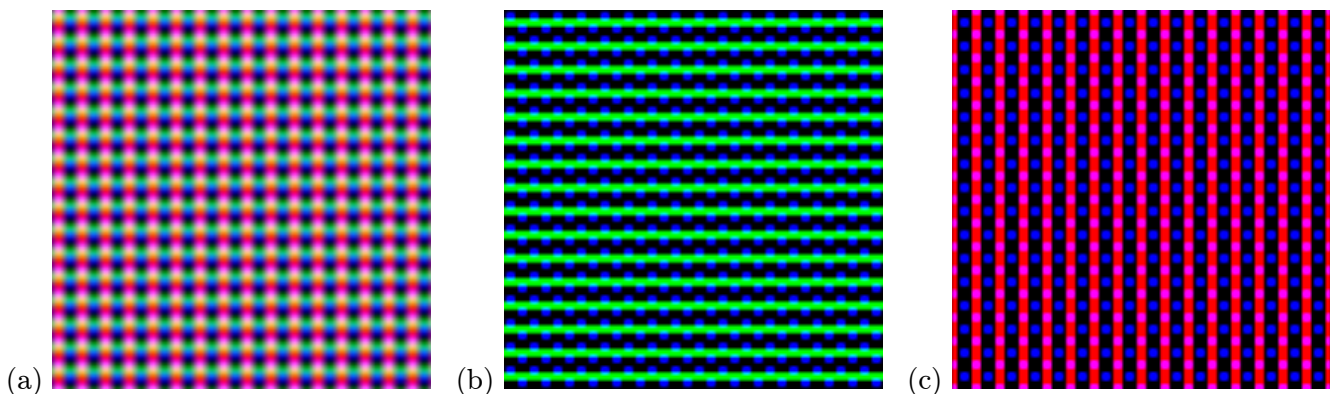


Figura 2: (a) Textura original, filtradas (b) horizontal y (c) verticalmente.

3 Filtrado Espacial Adaptativo

La segunda idea estudiada es aplicar filtrado espacial adaptativo. 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 manera similar a la vista en la LIC con su integral lineal. El filtrado por convolución es capaz de producir excelentes resultados aún con *kernels* de tamaño reducido (3×3), lo que produce el efecto esperado a un bajo costo computacional. El filtrado para resaltar bordes es un ejemplo de filtrado direccional. En la Fig. 2 podemos observar una textura filtrada horizontal y verticalmente aplicando filtrado de Sobel [11, 6]. 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 las direcciones 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 localmente rotado en la dirección que en cada punto asume el campo vectorial, y filtrar la textura en ese lugar con dicho filtro local. En la Fig. 3 podemos ver algunos ejemplos, donde la misma textura es convolucionada respectivamente con los siguientes filtros:

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

Se puede observar que el segundo kernel no produce discontinuidades en la visualización, debido a su simetría. 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 iteración entre este filtrado direccional adaptativo y los métodos de texturas ya vistos es bastante sencilla e inmediata. Esta 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 (llamada doble CLIC), 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 cada una de ellas iterada independientemente de la otra no produce cambios, pero sí lo hace su alternancia, debido a

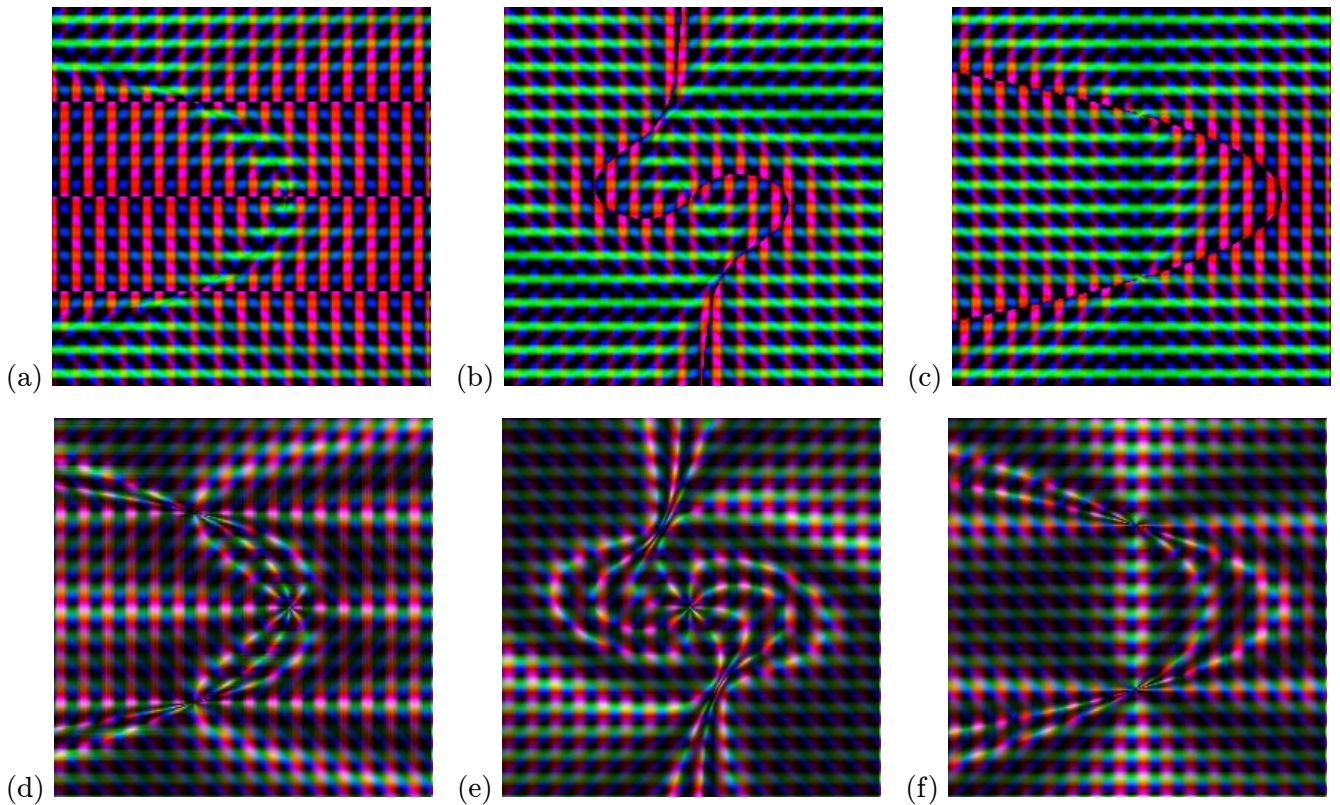


Figura 3: Dos sistemas convolucionados con *Filtro 1* (*a*, *b* y *c*), y *Filtro 2* (*d*, *e* y *f*).

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. En la Fig. 4, se muestra el resultado de aplicar CF con un filtro simétrico como el de Sobel. 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 Trabajo en Desarrollo

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 utilizar un *kernel* 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, y utilizar el mapa de direcciones e intensidades, rotándolo según la dirección local del sistema para codificar

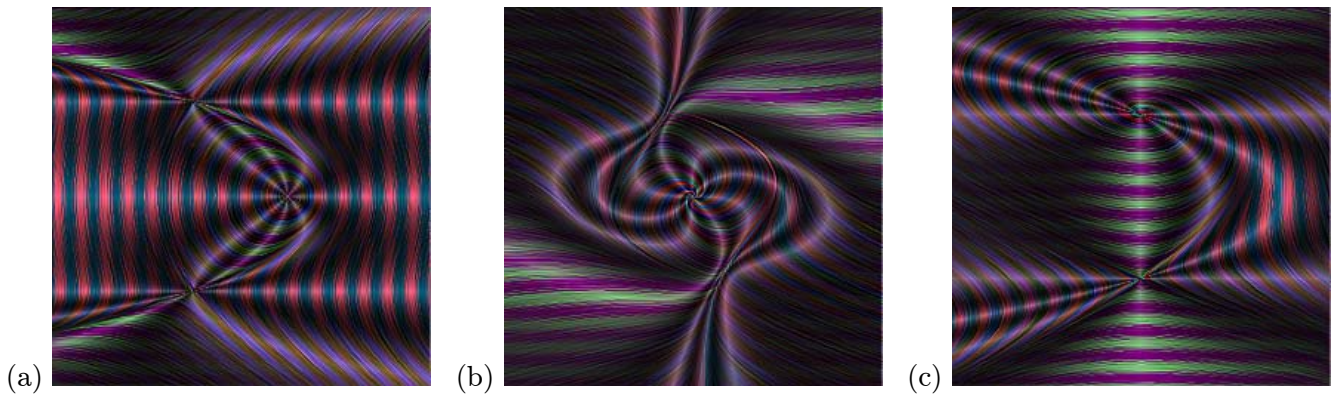


Figura 4: CF aplicada a los mismos sistemas que en la Fig. 3.

la dirección como un filtrado direccional, donde 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 Hypersstreamlines. *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, Julián Dominguez, and Andrés Repetto. Advanced Techniques for Real-time Flow Visualization. In *SPIE Proceedings Vol. 4716*, pages 375–385, The Intl. 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] J. Gomes and L. Velho. *Image Processing for Computer Graphics*. Springer, New York, 1997.
- [8] R. González and R. Woods. *Digital Image Processing*. Addison-Wesley, 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 Strogatz. *Nonlinear Dynamics and Chaos*. Addison-Wesley, Reading, 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.