

MÉTODOS INTERACTIVOS PARA LA VISUALIZACIÓN DE FLUJOS

Julián Dominguez, Claudio Delrieux y Andrés Repetto

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

Palabras Clave: Visualización Científica — Flujos y sistemas dinámicos.

Abstract. *La Visualización Científica se ha convertido en una de las herramientas más importantes en el estudio de la dinámica de sistemas no lineales, análisis numérico de espacios vectoriales, y el estudio de flujos y mecánica del continuo. En la mayoría de los casos particulares de estos sistemas, las soluciones analíticas cerradas son excesivamente complejas o bien imposibles de encontrar. Por dicha razón el análisis numérico y la interpretación gráfica de los resultados representan la única solución práctica posible. Los métodos usuales para la visualización de diagramas de fase y espacios vectoriales, así como las demás técnicas de análisis en sistemas dinámicos y mecánica del continuo, parecen generar una disyuntiva de calidad vs. interactividad. En efecto, las técnicas que proveen posibilidades de interactividad (vectores o streamlines, por ejemplo) se caracterizan por producir resultados de calidad pobre, lo cual puede llevar en muchos casos a una interpretación errónea de los resultados visuales. Otras técnicas que producen visualizaciones de mayor calidad, como por ejemplo la LIC (Line Integral Convolution) producen resultados gráficos adecuados sin requerir intervención, pero con un costo computacional excesivo, aún en gráficos de bajas resoluciones. En este trabajo presentamos un una nueva técnica que permiten alcanzar resultados que combinan calidad, riqueza de información y tiempos interactivos. La misma se basa en la aplicación local de un filtro direccional sobre una textura, tomando como dirección la correspondiente al espacio vectorial en cada punto. Esta técnica puede aplicarse en forma conjunta con las existentes, produciendo resultados de muy buena calidad con tiempos interactivos.*

1 INTRODUCCIÓN

La visualización científica ha cobrado importancia con el paso de los años, debido principalmente al creciente desarrollo de la tecnología, lo cual posibilita el cómputo numérico de grandes volúmenes de datos en tiempos interactivos, aún en computadoras personales. En consecuencia, se ha tornado más sencillo para investigadores de todo tipo de disciplinas el desarrollo de modelos matemáticos y su análisis y simulación numérica. Esto hace cada vez más ventajoso el uso de métodos numéricos que intentar resolverlo analíticamente, lo cual muchas veces ha probado ser de una altísima complejidad, si no imposible en la mayoría de los casos.

Pero el análisis numérico de un modelo no sería posible si no existiera una forma de conocer y comprender los resultados relevantes. La visualización de determinadas propiedades del sistema nos ayuda a interpretar en una forma rápida e intuitiva su comportamiento para una condición dada. Esta idea conceptual debe darnos la mayor cantidad de información posible acerca del sistema, a la vez de proporcionarnos la suficiente flexibilidad para permitir su análisis en forma interactiva. El objetivo del presente trabajo consiste en lograr una metodología de visualización del comportamiento dinámico de sistemas que sea adecuada para el correcto estudio de estos sistemas, y de un relativamente bajo costo computacional y manejo sencillo e intuitivo que permita tiempos interactivos con el usuario.

La visualización aplicada al estudio de estos temas consiste en obtener una idea conceptual del comportamiento de un sistema dinámico a partir de sus ecuaciones diferenciales. Esto ha llevado a la creación de numerosas técnicas y metáforas visuales con el fin de caracterizar el comportamiento del sistema de la manera lo más fiel posible, y a partir de cualquier condición inicial. De esa manera fueron propuestos métodos tales como glifos, trayectorias de partículas, íconos, Streamlines, Hyperstreamlines, Spot Noise, LIC, etc. [1, 2, 3, 10, 15, 16] Estas técnicas 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 presentamos una técnica que permiten alcanzar resultados que combinan calidad, riqueza de información y tiempos interactivos. En la siguiente Sección se muestran brevemente los métodos tradicionales (flechas y streamlines), en la Sec. 3 veremos la evolución respecto de los métodos basados en texturas e introducir algunos métodos mejorados presentados anteriormente, en particular la *Cumulative Line Integral Convolution*. En la Sección siguiente presentamos un nuevo método, que se basa en la aplicación local de un filtro direccional sobre una textura, tomando como dirección la correspondiente al espacio vectorial en cada punto. A lo largo de todas las Secciones utilizamos dos ejemplos de sistemas, para poder comparar adecuadamente los resultados obtenidos.

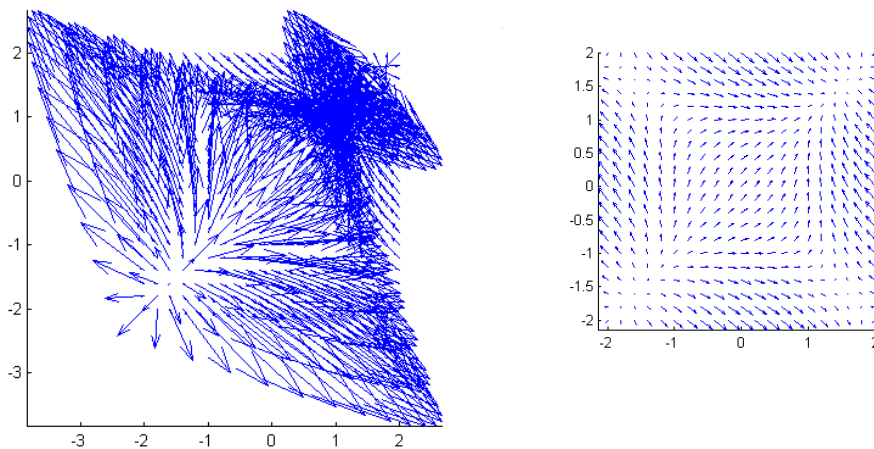


Figura 1: El sistema (1) representado con flechas, (a) sin normalizar, y (b) normalizado.

2 MÉTODOS CONVENCIONALES

La visualización de campos vectoriales se inició tradicionalmente con la más sencilla y directa manera de representar un vector: una flecha. Esta metáfora codifica los vectores tangentes al campo en sentido y magnitud, proveyendo los datos necesarios para representar las magnitudes principales asociadas a un vector. La primer complicación que presenta este tipo de representación es la necesidad de normalizar la magnitud de los vectores, para poder visualizar todo el campo vectorial y no producir solapamientos. Consideremos el siguiente sistema dinámico:

$$\begin{cases} \dot{x} = 2\cos(x) - \cos(y) \\ \dot{y} = 2\cos(y) - \cos(x) \end{cases} \quad (1)$$

En la Fig. 1 se muestran los resultados obtenidos al representar con flechas las ecuaciones del sistema (1), graficándolas con una herramienta matemática muy popular para este tipo de análisis. Este sistema tiene tres diferentes tipos de puntos críticos: fuentes o repeledores, sumideros o atractores, y puntos silla. La identificación visual de ellos es realmente deficiente.

A causa de estas deficiencias, surge como idea natural la de graficar curvas integrales (streamlines) en el espacio de fases. La representación de un diagrama de fases con streamlines se basa en elegir una adecuada cantidad de condiciones iniciales (semillas), y a partir de cada una de ellas realizar una integración numérica. Los parámetros de esta visualización son entonces la cantidad y posicionamiento de las semillas, la cantidad de pasos de integración a realizar en cada streamline, y el valor del diferencial finito de integración. Esta técnica, así como la anterior, tiene sus propias ventajas y desventajas. También es rápida como las flechas (dependiendo de la longitud de las streamlines), pero supera a aquella en no producir solapamiento (y por lo tanto no necesitar de normalización ninguna). Sin embargo permanece el problema de un cubrimiento no uniforme del dia-

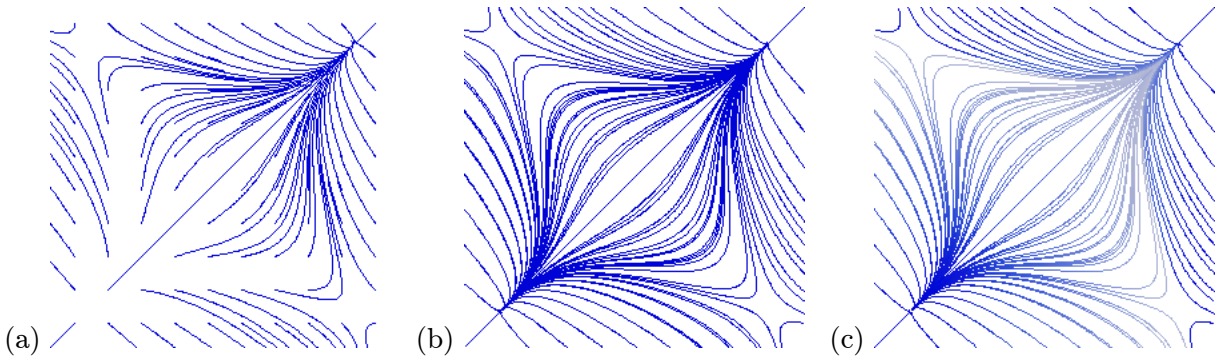


Figura 2: El sistema (1) representado por: (a) Streamlines, (b) Streamlines + Path Search, y (c) Streamlines + Path Search con desaturación.

grama de fases. A la vez se pueden producir interpretaciones ambiguas al no visualizarse explícitamente el sentido del flujo. Por último, determinadas situaciones como por ejemplo trayectorias homoclínicas, o ciclos límite metaestables, son difícilmente representados de manera clara.

Algunas de estas limitaciones se pueden solucionar con técnicas más o menos elaboradas. Por ejemplo, la dirección de la dinámica puede representarse modificando la saturación del color de la representación del streamline. De esa manera, los atractores se visualizan con colores menos saturados, y los repeledores con color fuerte. También puede mejorarse el despaseo cubrimiento debido a la estabilidad local por medio de un mecanismo de búsqueda inversa o recorrido “hacia atrás” del streamline, de manera que la graficación del mismo no comience exactamente en la posición de la semilla, sino que se realice una dinámica inversa previa. En la Fig. 2 puede verse el mismo sistema dinámico que en la Fig. 1, pero representado con estas técnicas.

Una comparación final puede realizarse con el siguiente sistema:

$$\begin{cases} \dot{x} = x - y - x \cdot (x^2 + 5 \cdot y^2) \\ \dot{y} = x + y - y \cdot (x^2 + 5 \cdot y^2) \end{cases} \quad (2)$$

En este caso el sistema tiene un ciclo límite, el cual es imposible de apreciar con flechas o con streamlines básicos (ver Fig. 3). Para tener en cuenta en cuanto a la interactividad alcanzable con estos métodos, en la máquina implementada (PIII 800MHz) los tiempos estuvieron en el orden de 200mseg para las flechas, 300-400mseg para los streamlines, y en los 500-600mseg para los streamlines con Path Search y desaturación.

3 MÉTODOS BASADOS EN TEXTURAS

La idea básica de los métodos basados en texturas es la de realizar una convolución local entre una textura de entrada y un segmento de trayectoria dentro del diagrama de fases. El mecanismo mas simple de estos es llevado a cabo a través de las tangentes del campo vectorial, y los más actuales incluyen una convolución de línea, como la ideada por Cabral

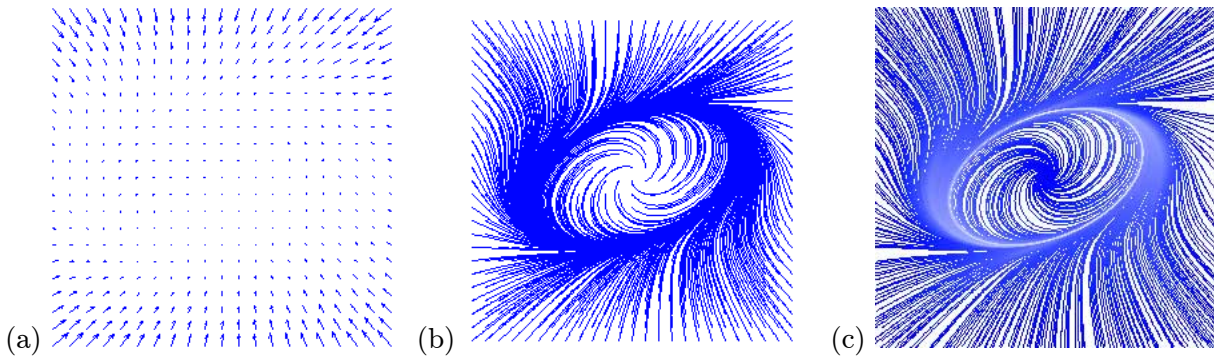


Figura 3: El sistema (2) representado por: (a) Flechas, (b) Streamlines, y (c) Streamlines + Path Search + desaturación, donde se puede apreciar claramente el ciclo límite del sistema.

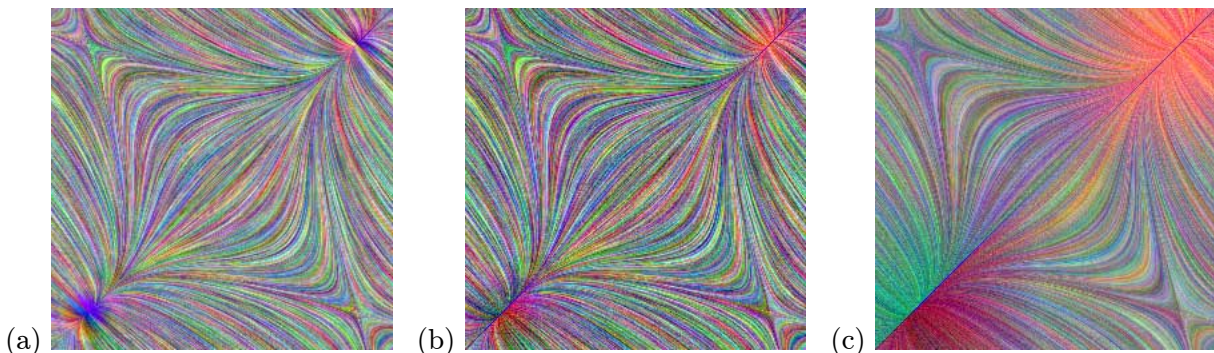


Figura 4: (a) Sistema (1) visualizado con LIC y $L = 30$ ($t = 16$ seg.), (b) con paso adaptativo y $L = 30$ ($t = 8.4$ seg.), y (c) paso adaptativo y $L = 100$ ($t = 26$ seg.).

y Leedom en 1993 [1] conocida como Line Integral Convolution, o *LIC*. El efecto de esta representación es visualizado como un sfumado en la dirección del flujo.

Por la metodología empleada, la LIC se hace atractiva para la visualización de sistemas cuyo diagrama de fase es desconocido, ya que el resultado no depende en nada de los parámetros de la evaluación (cantidad y longitud de trayectorias, disposición de las semillas, etc.). Tampoco se corre el riesgo de que las zonas inestables del diagrama de fases queden raramente cubiertas. Por otro lado, la visualización del flujo es continua y las principales características del sistema quedan al descubierto. La calidad del LIC depende de la longitud L (en pixels) de la curva convolucionada, o tamaño del *kernel*. También puede ser importante el uso de una función ponderadora que asigne mayor o menor importancia a los diferentes pixels de la textura. En [14, 9] se pueden encontrar más información acerca de las curvas integrales, el teorema de convolución, y la influencia de la forma del kernel en la LIC. En las Fig. 4(a) se presenta el sistema (1) graficado con esta técnica. Se observa que la calidad de las representaciones es muy buena, pero al mismo tiempo los costos computacionales son muy altos.

Una optimización que mejora en mucho la performance de este algoritmo consiste en modificar adaptativamente el paso de integración en función del valor local de la velocidad

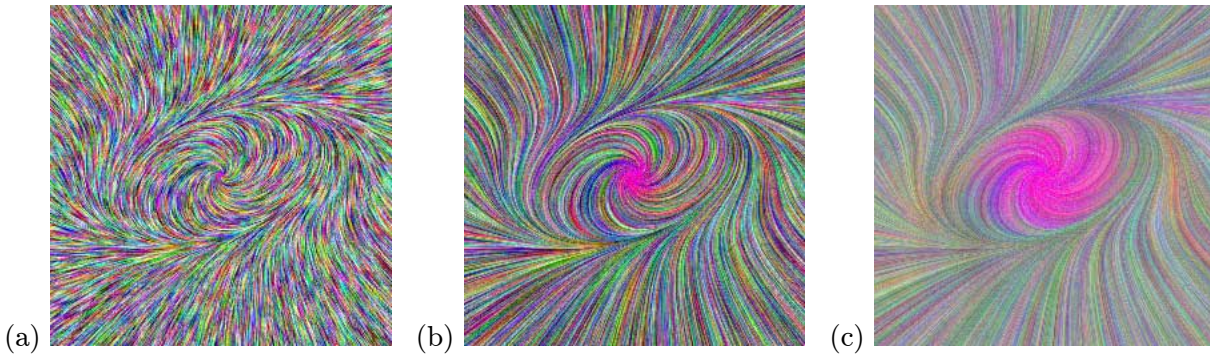


Figura 5: El sistema (2) representado por el algoritmo LIC de paso adaptativo. (a) $L = 5$ y $t = 0.8$ seg., (b) $L = 30$, y $t = 4.2$ seg., y (c) $L = 100$ y $t = 10.6$ seg.

de la trayectoria. Esta idea no solo hace más eficiente la evaluación, dado que evita el sobremuestreo, sino que también puede mejorar la calidad al evitar también la posibilidad de submuestreo. En las Figs. 4 y 5 se puede observar el resultado de aplicar esta técnica y la mejora en los tiempos de cómputo. Se aprecia que el resultado es más suave a mayor longitud de kernel, también aumenta mucho el costo computacional con kernels grandes, y además no es fácil interpretar el gráfico para determinar la existencia o no de una trayectoria cerrada. Existen otras técnicas originadas a partir de la LIC, como la de Spot Noise [15] y la de Ink Droplets [12], que pueden implementarse con una mayor velocidad de cómputo, pero no producen necesariamente resultados de buena calidad.

Sin embargo, una mejora considerable puede obtenerse a partir de una modificación de la LIC, en la cual no se realiza una convolución completa de todos los pixels de la textura visitados por la trayectoria, sino que se utiliza un acumulador para computar el color final a partir de los pixels visitados. Este método, denominado *CLIC* o *Cumulative Line Integral Convolution* fue propuesto y estudiado en trabajos anteriores [5, 4], y muestra reunir las ventajas de streamlines y LIC en forma simultánea. En la Fig. 6, se pueden comparar los tiempos y resultados que este método produce, observándose su satisfactoria calidad y de bajo costo computacional. La representación genera un cubrimiento uniforme del diagrama de fases y los puntos críticos del sistema son adecuadamente representados.

En trabajos posteriores se presentaron variantes y mejoras al método, como el *Doble CLIC*, consistente en una pre-recorrida del campo en sentido inverso, para luego convolucionarlo en el sentido directo, lo cual produce resultados más suaves. También existen variantes de menor tiempo de cómputo, que aprovecha los resultados intermedios obtenidos anteriormente (llamada variante *R* del método), pero a la vez produce una pérdida en la precisión de los cálculos.

Un tema de gran importancia pero que no ha recibido suficiente atención es el de la textura de entrada para el algoritmo LIC y sus variantes. Hasta el momento hemos presentado todos los métodos basados en texturas graficados con una textura de entrada de tipo ruido blanco, que es la textura clásica utilizada para ejemplificar estos métodos.

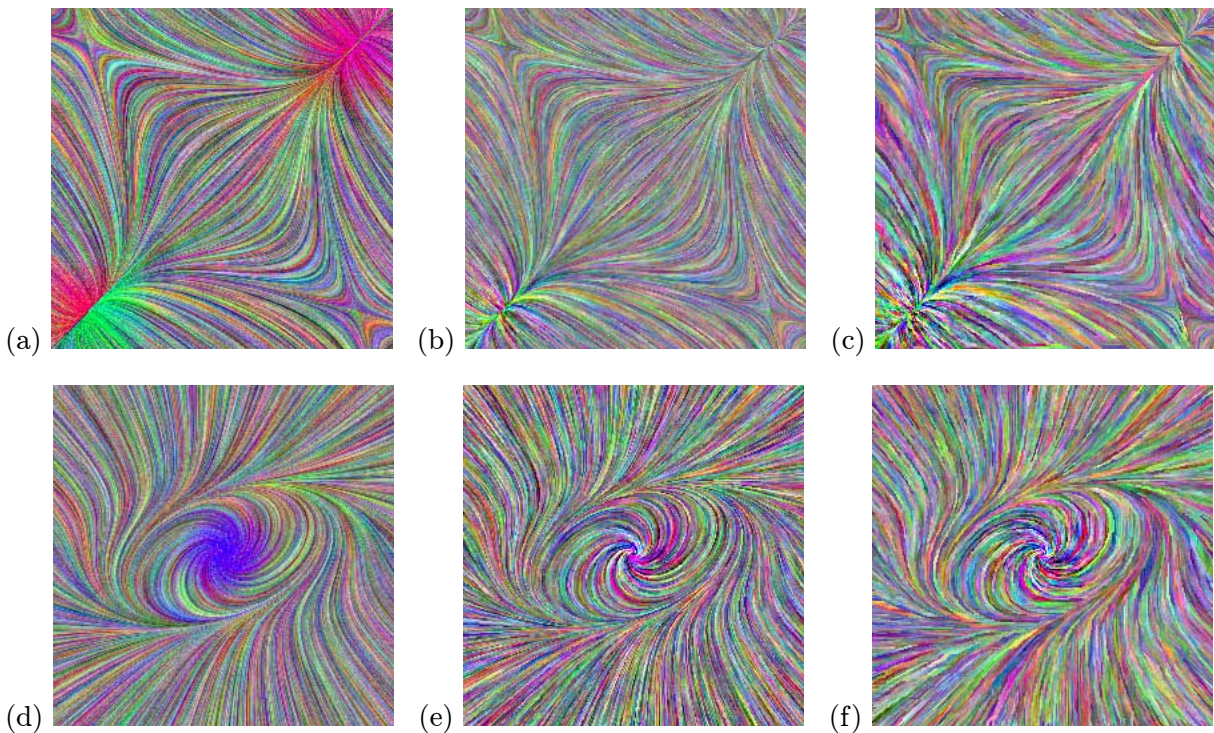


Figura 6: Ambos sistemas analizados, teniendo en (a) y (d) el sistema graficado por LIC adaptativo con un kernel de 50, arrojando tiempos de 13seg (a) y 6seg (d). Luego en (b) y (e) el método CLIC, con kernel de 500 y tiempo de aproximadamente 1.5seg. Finalmente, en (c) y (f) las versiones tipo *R* del método CLIC, con tiempos del orden de los 800mseg.

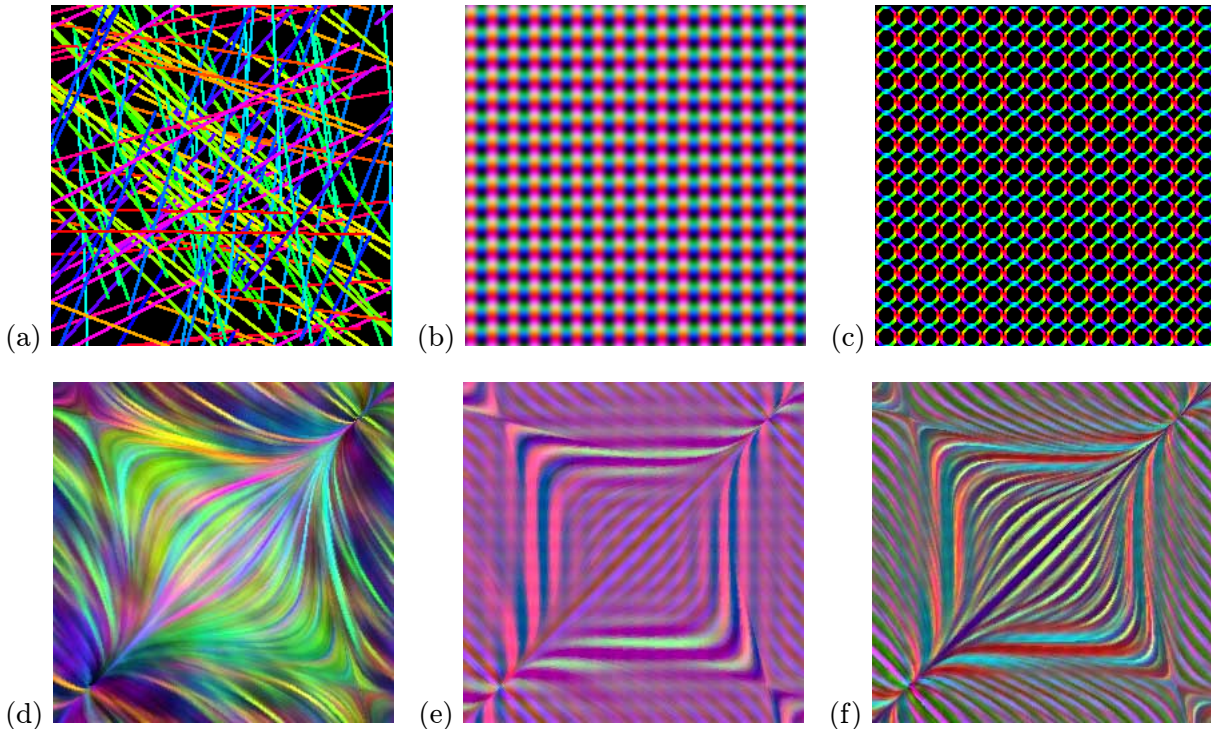


Figura 7: Tres texturas diferentes, donde todas tienen codificado en sus colores la dirección, y los resultados de convolucionarlas en el mismo sistema (1) utilizando el algoritmo Doble CLIC. En todos los casos el tiempo de ejecución está en el orden de los 2 seg.

Sin embargo, hemos destacado también la importancia que tiene la composición de la textura para resaltar las características del sistema en este tipo de métodos. 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, interaccionan con la textura, realizando un esfumado direccional en el sentido del campo vectorial.

La importancia y utilidad de este aspecto para la eficacia de la visualización ha sido poco estudiado (con excepción de [11] 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.

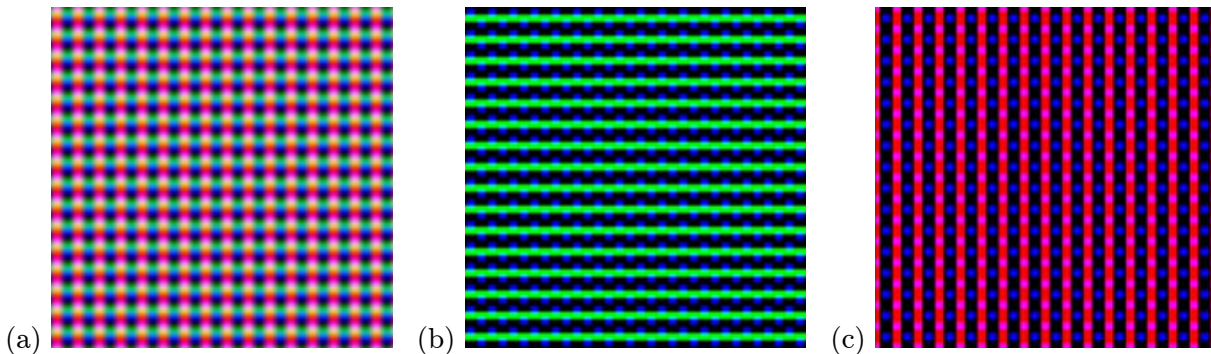


Figura 8: A partir de *a*) obtenemos la texturas filtradas horizontal (figura *b*, campo a 0°) y verticalmente (figura *c*, campo a 90°).

4 FILTRADO ADAPTATIVO DIRECCIONAL

Como pudimos ver en la Sección anterior, la LIC resalta las características de la textura cuya dirección coincide con el campo en dicho punto. 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 y video, segmentación y reconocimiento de formas, comunicaciones, etc.

El filtrado por convolución es capaz de producir excelentes resultados aún con kernels pequeños (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, mostramos como un *simple* kernel estático nos permite visualizar una dirección determinada que se encuentra *en fase* con la textura de entrada. En este caso es un filtrado horizontal (o vertical si rotamos 90° la matriz o *kernel*), denominado filtro de Sobel. [13, 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. 8 podemos observar una textura filtrada horizontal y verticalmente. 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). En el caso de la visualización de flujos, el interés radica en resaltar la dirección del mismo en cada punto dentro del diagrama de fase. Una forma de lograrlo sería aplicar un filtrado adaptativo, cuya dirección esté de acuerdo a la de la dinámica local del sistema.

Así como la LIC produce una convolución lineal sobre las trayectorias, el *Filtrado Adaptativo Dinámico* (o *FAD*) produce una convolución espacial en el entorno del punto

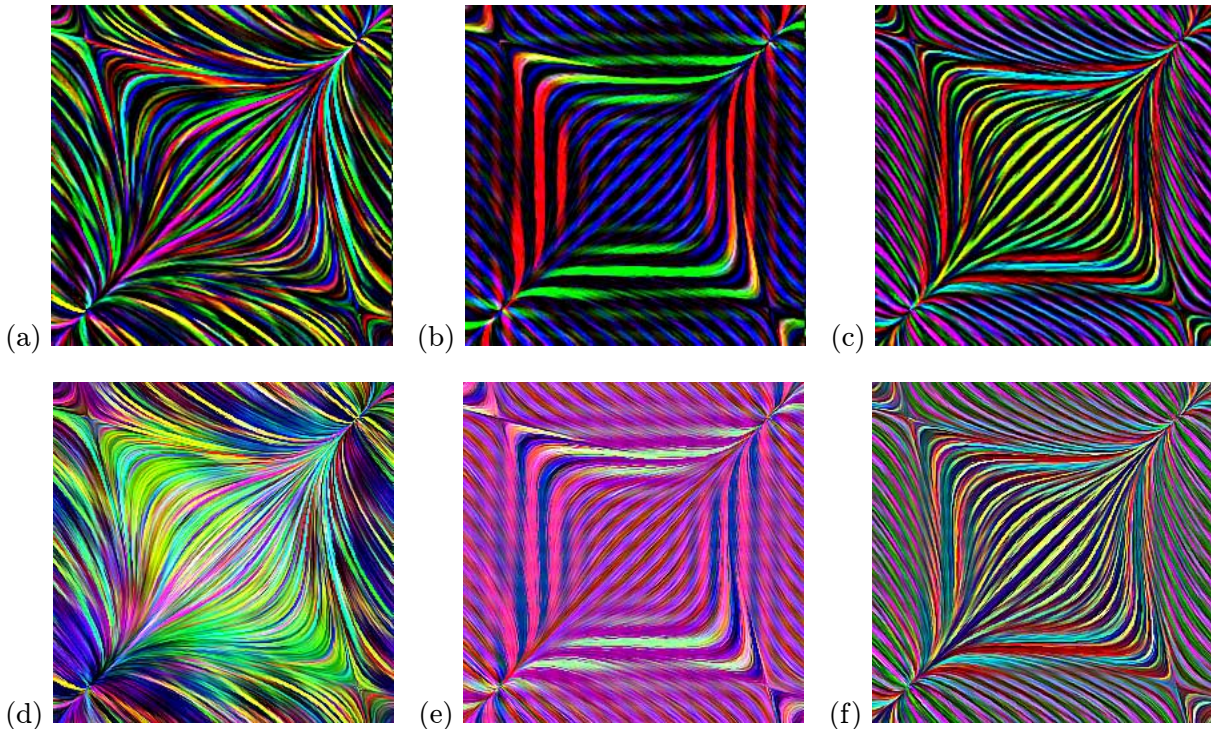


Figura 9: Dos secuencias de los mismos sistemas convolucionados con el *Filtro 1* (a, b y c), y el *Filtro 2* (d, e y f). En todos los casos el tiempo de ejecución ronda en los 350mseg.

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 una rotación local en cada punto (en función de la dirección local del flujo en dicho punto), y filtrar en ese lugar la textura para resaltar la dinámica local. En la Fig. 9 vemos como, los mismos sistemas graficados en la Fig. 7, ahora mejorados con los siguientes dos filtros de resalte horizontal:

$$Filtro1 : \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}, \quad Filtro2 : \begin{pmatrix} -1 & -1 & -1 \\ 2 & 3 & 2 \\ -1 & -1 & -1 \end{pmatrix}$$

Con esta técnica se consiguen resultados de muy buena calidad, mejorando los métodos ya existentes basados en texturas como la LIC y el CLIC. De esta forma, hemos conseguido el resultado esperado: resultados de alta calidad en tiempos interactivos, lo que los convierte en una herramienta poderosa para la interpretación de resultados numéricos.

5 CONCLUSIONES Y TRABAJO FUTURO

En este trabajo brindamos una breve revisión de los métodos clásicos de visualización, avanzamos sobre las técnicas más modernas, para llegar a proponer mejoras que permiten hacer de estos métodos herramientas de visualización muy poderosas. A la vez, presentamos un nuevo método que puede reforzar la eficacia obtenida por otros métodos basados en texturas como LIC o CLIC. Su potencia se halla en realizar un análisis en un entorno del punto analizado, diferenciándolo de la LIC que sólo tiene en cuenta la dirección del campo vectorial.

Con estas técnicas combinadas o de forma independiente, el investigador tendrá a su alcance la posibilidad de manipular los parámetros de ajuste para lograr entender o modelar un dado problema. El bajo costo de las mejoras introducidas hacen que a la vez se conviertan en herramientas interactivas, permitiendo un fácil estudio del sistema en análisis.

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 Hypersurstreamlines. *IEEE Computer Graphics and Applications*, 13(4):25–33, 1993.
- [4] Claudio Delrieux, Julián Domínguez, and Andrés Repetto. Mejorando la Visualización de Campos Vectoriales. In *VI ICIE*, pages 397–416, International Conference on Information Engineering, 2001.
- [5] Claudio Delrieux, Julián Domínguez, 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.
- [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 González and Richard Woods. *Digital Image Processing*. Addison-Wesley, Wilmington, USA, 1996.
- [9] Hans-Christian Hege and Detlev Stalling. Fast LIC with Piecewise Polynomial Filter Kernels. In H.-C. Hege and K. Polthier, editors, *Mathematical Visualization*, pages 295–314. Springer-Verlag, Amsterdam, 1998.
- [10] James Helman and Lambertus Hesselink. Visualizing Vector Field Topology in Fluid Flows. *IEEE Computer Graphics and Applications*, 11(3):36–46, 1991.

- [11] 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.
- [12] H. Löffelmann, A. König, and E. Gröller. Fast Visualization of 2D Dynamical Systems by the Use of Virtual Ink Droplets. In *Proceedings of Spring Conference on Computer Graphics*, pages 111–118, Slovakia, 1997. Eurographics.
- [13] J. C. Russ. *The Image Processing Handbook*. CRC Press, Boca Raton, FL, 1989.
- [14] D. Stalling and H.-C. Hege. Fast and Resolution Independent Line Integral Convolution. In *SIGGRAPH 95 Conference Proceedings*, pages 249–256, Los Alamitos, CA, 1995. ACM SIGGRAPH, ACM PRESS.
- [15] J. J. van Wijk. Spot Noise: Texture Synthesis for Data Visualization. *ACM Computer Graphics*, 25(4):309–318, 1991.
- [16] J. J. van Wijk. Flow Visualization with Surface Particles. *IEEE Computer Graphics & Applications*, 13(7):18–24, 1993.