

Visualización Interactiva de Diagramas de Fase

Claudio Delrieux, Julián Dominguez y Andrés Repetto

Universidad Nacional del Sur, Alem 1253, (8000) Bahía Blanca, ARGENTINA.
Voice: (54)(291)4591501 ext. 3381 — Fax: (54)(291)4595154 — e-mail: claudio@acm.org

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

1 Introducción

En la literatura se han presentado varios métodos para la visualización de campos vectoriales (glifos, trayectorias de partículas, íconos, Streamlines, Hyperstreamlines, Spot Noise, LIC, etc.) [1, 2, 3, 12, 15, 16], de las cuales Streamlines y LIC son las más importantes y utilizadas. En flujos estacionarios, el método de las Streamlines o líneas de flujo coincide con la integración de trayectorias en sistemas dinámicos [9, 14]: se posiciona una condición inicial (o *semilla*) en el espacio de fases, y se integra la ecuación diferencial por medio de diferenciales finitos. Para obtener un cubrimiento uniforme del diagrama de fases o para ubicar todos los puntos críticos del sistema, se requiere encontrar un conjunto representativo de trayectorias, lo cual implica encontrar las respectivas semillas, y el paso y la longitud de integración. La elección de un valor para el diferencial finito o factor de integración debe ser realizada cuidadosamente para no producir excesivo oversampling o undersampling. La única manera de asegurar un paso óptimo en toda circunstancia consiste en utilizar un paso adaptativo, como se sugiere en [4]. La cantidad de iteraciones, junto con el paso de integración y la velocidad media en la trayectoria, son todos factores que inciden en la longitud final que tendrá la misma en el diagrama de fases. En muchas circunstancias la correcta apreciación de un campo de velocidades se logra visualizando un gran conjunto de trayectorias relativamente cortas (“colas de cometa” [6, 7]), y para brindar una buena idea de la dirección del flujo se pueden representar las trayectorias modificando la saturación [8]. El posicionamiento uniforme de las semillas no garantiza un cubrimiento satisfactorio. Ni siquiera garantiza que todos los puntos críticos del sistema dinámico se representen adecuadamente. Una solución para estos problemas consiste en realizar la dinámica inversa del sistema a partir de la semilla, una cierta cantidad de iteraciones, para determinar si el punto está cerca de un repeledor. Esta técnica de “path-search” [5] permite lograr un cubrimiento más parejo del diagrama de fases, y además acelera la generación del diagrama de fases porque requiere integrar una menor cantidad de trayectorias. Un problema inevitable es que muchas de estas técnicas son directamente dependientes del sistema en particular, por lo que una visualización adecuada requiere de la asistencia del usuario.

Por su parte, la LIC (Line Integral Convolution) [1] realiza una integración más corta (típicamente se realizan menos de 100 iteraciones). La trayectoria generada no se grafica, sino que se utiliza para recorrer los texels correspondientes, integrando el color de los mismos para

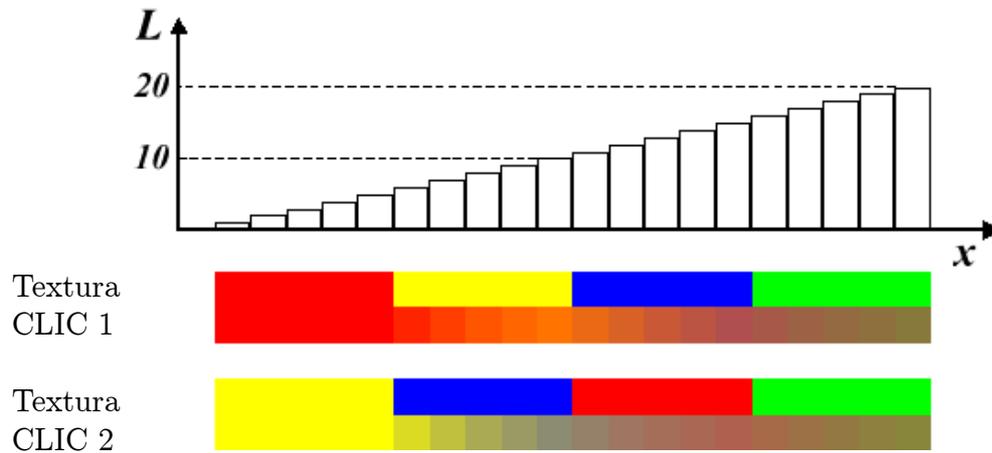


Figura 1: Representación gráfica (1D) del comportamiento del CLIC, aplicado a dos texturas. El sentido del flujo es de izquierda a derecha.

encontrar el color final del pixel. Por dicha razón, la LIC produce un cubrimiento uniforme del diagrama de fases, por lo que todos los puntos críticos del sistema son satisfactoriamente representados. Una mejora inmediata consiste en multiplicar, para cada pixel, el aporte de cada texel por medio de una misma función ponderadora o *kernel* de convolución de longitud L igual a la cantidad de pasos de integración. Sin embargo muchas veces resulta difícil identificarlos adecuadamente. No siempre se pueden distinguir atractores de repelidores, y resulta casi imposible determinar la existencia de trayectorias cerradas (focos, ciclos límite, trayectorias homoclínicas). La mejora de las posibilidades de la LIC constituye un activo campo de investigación [8, 10, 11, 13, 17], especialmente en lo que concierne a mejorar su costo computacional y la representación del sentido del flujo y las trayectorias cerradas. Muchas de las mejoras mencionadas más arriba para las streamlines son directamente aplicables en la LIC. En particular podemos mencionar que la LIC adaptativa es aproximadamente un orden de magnitud más veloz [4]. Por último, podemos mencionar un aspecto de importancia: la textura utilizada para la generación de la LIC. 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. En [5, 6] se sugiere un conjunto de técnicas para generar texturas más adecuadas, las cuales luego pueden ser utilizadas para resaltar diversos aspectos de un determinado campo vectorial.

2 LIC acumulativo

El LIC acumulativo (*Cumulative LIC* o *CLIC*) [5, 6] surge de combinar las técnicas de desaturación de las trayectorias, y de posicionamiento no uniforme de las semillas en las streamlines. La idea consiste en acumular el color de los texels visitados en una LIC, en vez de computar una convolución (ver Fig. 1), e ir graficando el color acumulado en la imagen final. De esa forma, el cómputo del color final de un pixel es mucho más sencillo que con la LIC, pueden reutilizarse los cálculos parciales para encontrar los colores de los pixels vecinos, y cada trayectoria computada genera varios pixels en la imagen final. En general esta técnica permite bajar en promedio un orden de magnitud el tiempo de cómputo con calidades similares a la LIC original. Además el CLIC puede utilizarse con todas las demás

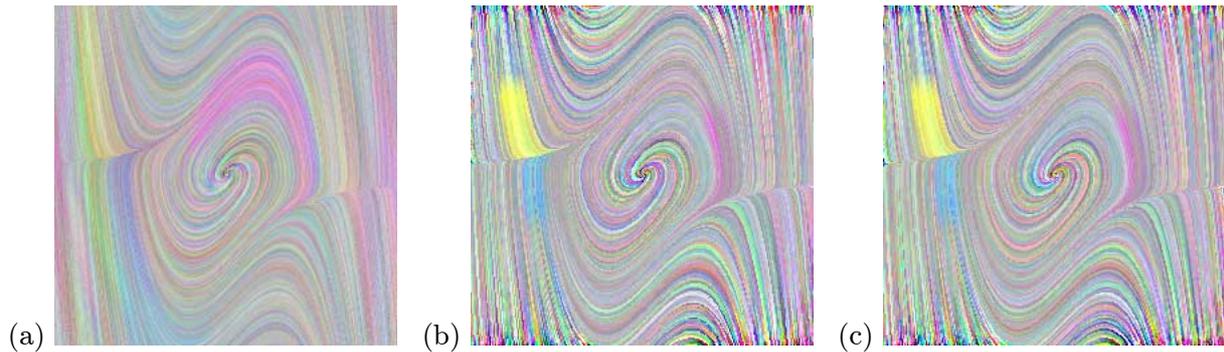


Figura 2: (a) LIC original, $L=300$, $t=212$ seg. (b) CLIC adaptativo sembrado en cada pixel, $L=300$, $t=2.5$ seg. (c) CLIC adaptativo con *path search*, sembrado al 10% y con segundo paso, $L=300$, $t=0.65$ seg.

técnicas mencionadas en la sección anterior (paso adaptativo, *path search*), con lo cual se obtiene un método cuyo tiempo de cómputo es muy cercano a los streamlines, pero con la calidad de la LIC. Un detalle importante del CLIC es que –manejado cuidadosamente– permite utilizar kernels mucho mayores que la LIC convencional (> 500) sin un incremento dramático en el tiempo de cómputo. Aplicado en forma directa, sobre cada pixel de la figura final, produce resultados similares a una LIC de mismo kernel L , con tiempos sensiblemente menores (ver Fig. 2). El mejor aprovechamiento de la técnica consiste en efectuar un “sembrado” no uniforme en un determinado porcentaje de los pixels de la imagen de salida. Experimentalmente podemos mencionar que generando una trayectoria CLIC cada 5 o 10 pixels, se produce un cubrimiento casi total de la imagen de salida (dependiendo de cómo sea el diagrama de fases y del valor de L elegido). El cubrimiento de la imagen final puede mejorarse sensiblemente utilizando *path search*, y eventualmente se puede computar un segundo paso de CLIC para generar una trayectoria en cada pixel no visitado, obteniéndose de esa manera un cubrimiento total. Es fácil ver, entonces, que aumentando la longitud L del kernel no aumenta necesariamente el tiempo de cómputo del CLIC. Un L mayor genera trayectorias más largas, con más pixels visitados, lo cual permite disminuir tanto la cantidad de trayectorias a recorrer como la cantidad de cómputo en el segundo paso.

Una observación detallada realizada a las imágenes resultantes puede mostrar que las trayectorias con L grandes van perdiendo saturación muy rápidamente, lo cual dificulta la visualización de las características geométricas del diagrama de fases. Una forma de solucionar este problema consiste en utilizar una función ponderadora lineal (de muy bajo costo computacional) que permita acumular el color a lo largo de la trayectoria con menor intensidad. De esa forma el factor k o *carry* permite controlar la desaturación de las trayectorias. En la Fig. 3 podemos ver el efecto del factor k sobre el CLIC. Se observa que con menor k la influencia de la textura original es mayor.

3 Doble CLIC

Como pudimos ver en la Sección anterior, el CLIC y sus mejoras asociadas nos permiten visualizar diagramas de fase en tiempos semejantes a las streamlines, con calidades semejantes a la LIC. Sin embargo, el método es nuevamente dependiente del sistema dinámico en particular y requiere la elección cuidadosa de los parámetros por parte del usuario para ser correctamente utilizado. Según se mencionara más arriba, la textura de entrada juega un rol

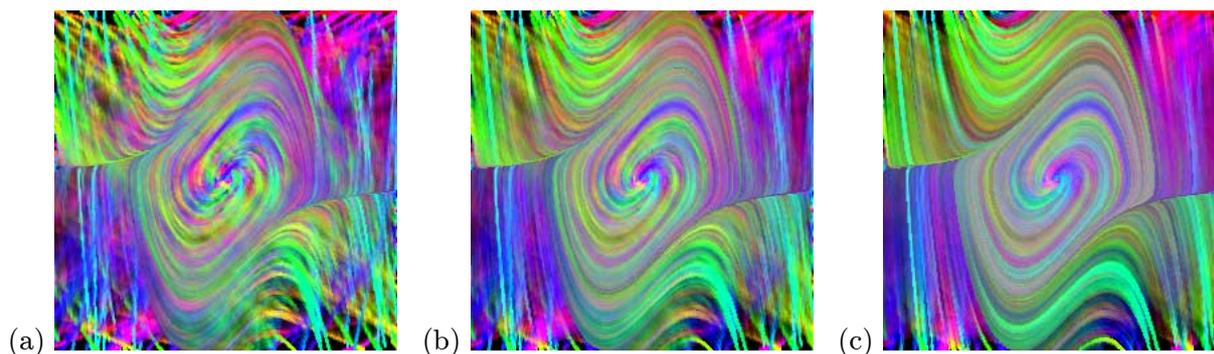


Figura 3: Visualizaciones del diagrama de fases del oscilador de Van der Pol (a) $k = 0.2$, (b) $k = 0.5$, y (c) $k = 1$.

preponderante en la calidad del resultado, y dado que el CLIC es notablemente económico para computar, la idea para lograr un método autónomo consiste en utilizar un CLIC (probablemente de baja calidad pero poco costo computacional) como *textura de entrada* para el cómputo del CLIC. Es decir, aplicamos dos veces el método. La primera transforma la textura original en una textura intermedia de mejor calidad, y la segunda produce la visualización final. De esa forma, el “doble CLIC” logra combinar la rapidez y autonomía de los streamlines con la calidad de la LIC. En la Fig. 4 podemos comparar, para un campo vectorial y una textura especialmente difícil, el efecto de aplicar dos veces el CLIC. De esa forma se logra un efecto visual de mayor semejanza al LIC original. Esta forma de trabajar permite elegir diferentes valores de k para realizar un CLIC inverso y luego hacia adelante (utilizando el primero como textura de entrada para el segundo). Se observa que la visualización producida por los nuevos métodos (CLIC y doble CLIC) son de calidad semejante a la LIC, con una eficiencia computacional mucho mayor.

Referencias

- [1] B. Cabral y L. Leedom. Imaging Vector Fields Using Line Integral Convolution. *ACM Computer Graphics (SIGGRAPH Proceedings)*, 25(3):263–270, 1993.
- [2] Roger Crawfis, Nelson Max, y 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. A Fast and Accurate Implementation of the Line Integral Convolution. En *CISST 2000 Conference*, págs 467–474, ISBN 1-892512-57-2, 2000.
- [5] Claudio Delrieux, Julián Dominguez, y Andrés Repetto. Mejorando la Visualización de Campos Vectoriales. In *VI ICIE*, págs. 397–416, 2001.
- [6] Claudio Delrieux, Julián Dominguez, y Andrés Repetto. Towards a CLIC in Vector Field Visualization. En *CISST 2001 Conference*, págs. 695–702, ISBN 1-892512-73-4, 2001.
- [7] Claudio Delrieux y Andrés Repetto. Visualización de Sistemas no Lineales y Caóticos. En *XXV CLEI '99*, págs. 241–253, 1999.

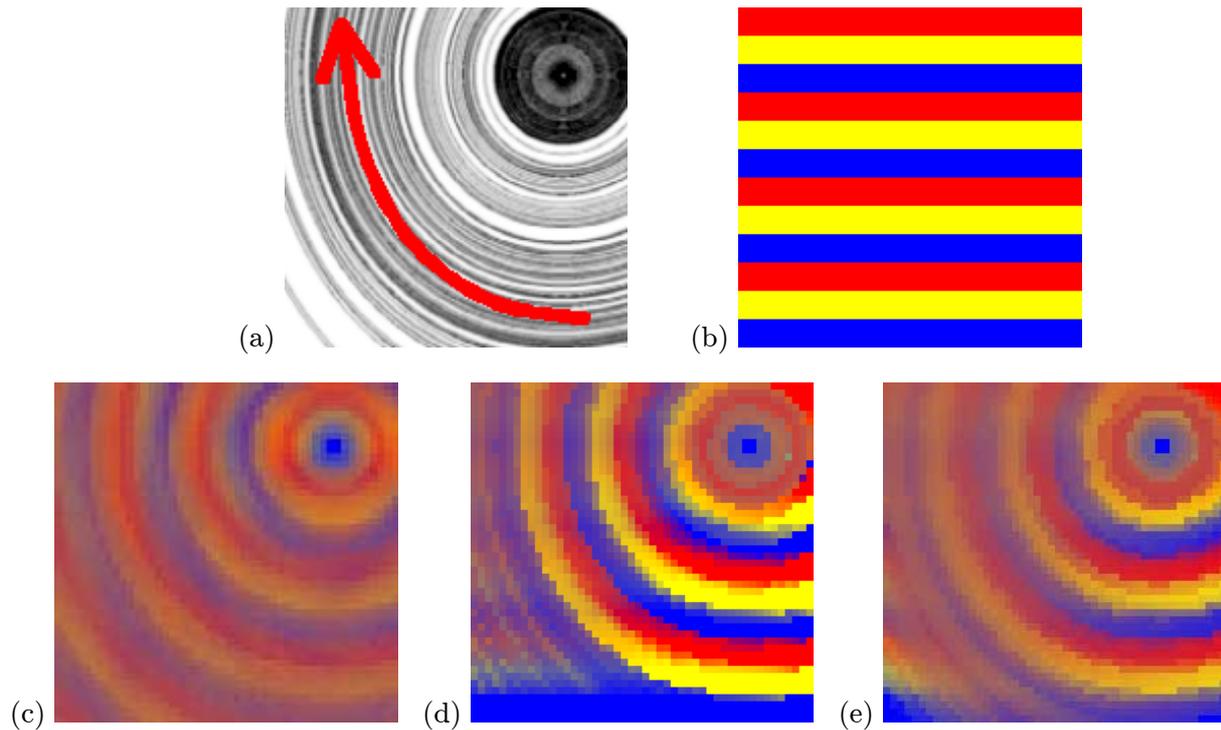


Figura 4: (a) Un campo vectorial, y (b) la textura utilizada. En (c) vemos la LIC respectiva, en (d) el CLIC y (e) el doble CLIC.

- [8] Claudio Delrieux y Andrés Repetto. Métodos para la Visualización de la Dinámica de Sistemas no Lineales. En *Argentine Symposium on Computing Technology*, págs. 153–162, 29 JAIIO, 2000.
- [9] Robert Devaney. *A First Course in Chaotic Dynamical Systems*. Addison Wesley, Reading, MA, 1992.
- [10] L.K. Forsell y S.D. Cohen. Using Line Integral Convolution for Flow Visualization: Curvilinear Grids, Variable-speed Animation, and Unsteady Flows. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):133–141, 1995.
- [11] Hans-Christian Hege y Detlev Stalling. Fast LIC with Piecewise Polynomial Filter Kernels. En *Mathematical Visualization*, págs. 295–314. Springer-Verlag, Amsterdam, 1998.
- [12] James Helman y Lambertus Hesselink. Visualizing Vector Field Topology in Fluid Flows. *IEEE Computer Graphics and Applications*, 11(3):36–46, 1991.
- [13] D. Stalling y H.-C. Hege. Fast and Resolution Independent Line Integral Convolution. En *SIGGRAPH 95 Conference Proceedings*, págs. 249–256.
- [14] Stefen H. Strogatz. *Nonlinear Dynamics and Chaos*. Addison-Wesley, Reading, Massachusetts, 1994.
- [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.
- [17] R. Wegenkittl, E. Gröller, y W. Purgathofer. Animating Flow-fields: Rendering of Oriented Line Integral Convolution. En *Computer Animation '97 Proceedings*, págs. 15–21.