

Medición del campo de velocidades en imágenes

Resumen

El análisis de la dinámica en imágenes es una de las componentes en el campo de la "Visión Artificial" (*Computer Vision*). El hecho de tener un trazador del campo de velocidades, el cual determina una matriz de las componentes de velocidades en 2D o 3D entre dos cuadros sucesivos de una toma de cámara de video, y además, que la resolución sea en tiempo real, nos brindarán una herramienta que en el futuro permitirá usarla en muchas diferentes aplicaciones. El presente trabajo es la primera etapa de una investigación mayor, la cual vinculará entre otros al "Estudio de Patrones en Imágenes (Faciales)". Este se centra en la aplicación de las ecuaciones de Lucas Kanade mediante la optimización de un algoritmo y su posterior verificación.

Palabras Claves

DINÁMICA - IMÁGENES - VISIÓN - ARTIFICIAL

Abstract

The analysis of dynamics in images is one of the components in the field of "Artificial Vision" (*Computer Vision*). The fact of having a tracer of the field of speeds, which determines a matrix of the components of speeds in 2D or 3D between two successive pictures of a taking of video camera, and that in addition the resolution is in real time,

will offer a tool that in future will allow to use it in many different applications.

The present work is the first stage of a greater investigation, which will tie among others to the "Study of Models in Images (Faces)". This one focuses the application of Lucas Kanade equations by means of an algorithm and its later verification.

Keywords

DYNAMICS - IMAGES - VISION - ARTIFICIAL

1. Introducción

El análisis de la dinámica entre cuadros de imagen obtenidos a partir de una cámara de video (objeto del presente trabajo), tiene gran importancia para su aplicación en varios campos de la ingeniería. El objetivo es lograr una herramienta que genere una matriz del campo de velocidades que represente la componente de esta en el plano (espacio), por cada uno de los píxeles de los cuadros en forma independiente (velocidad absoluta de cada píxel) o relativa a los puntos vecinos.

Para ello se centrará el desarrollo de la herramienta a la solución de sistemas complejos (con grados de libertad), como ser el campo de la fluido-dinámica. Se utilizará el método de "Lucas Kanade" el cual da lugar a un algoritmo que minimiza la complejidad computacional.

Por la extensión de este trabajo, se divide en tres etapas:

La primera etapa es el volcado del trazador seleccionado que es la ecuación de "Lucas Kanade" a un software que se desarrollará para ese fin y posteriormente verificar los resultados de los campos de velocidades obtenidos en cuadros de muestra, con los campos de velocidades obtenidos al procesar esos mismos cuadros por un software para aplicaciones de imágenes inteligentes.

La segunda y tercera etapa no es objeto del presente trabajo, pero se mencionan para que se conozca el objetivo final de toda la investigación. En la segunda etapa se trabajará con objetos pequeños (deben ser objetos sólidos como ser la filmación de un puñado de bolillas arrojadas al piso) moviéndose de manera independiente y fluidos (gases-líquidos) en movimiento, para el caso de fluidos, estos serán convenientemente tratados con un marcador para poder obtener imágenes del movimiento de las partículas. En la tercera etapa, se trabajará con placa para procesar gráficos (GPU) para hacer el procesamiento en paralelo y poder tener la resolución del campo de velocidades en tiempo real.

2. Desarrollo

2.1 Descripción

Los pasos a seguir para el logro del objetivo propuesto son:

- Separación y transformación de una toma de cámara de video, de los cuadros que la componen en cuadros de imagen en formato jpg (o cualquier otro estándar).
- Procesamiento de cada uno de esos cuadros de a dos a fin de poder obtener la matriz de velocidades de los objetos que se estén moviendo.
- Volcado de los valores obtenidos a un gráfico de vectores.
- Corroboración de los resultados obtenidos a fin de poder dar una validación primaria a la herramienta.

Como se ha mencionado, se debe tener acceso a cuadros de imagen con la condición de que el segundo sea consecutivo del primero; se busca un software libre que permita separar cuadros en formato de video a cuadros en formato jpg (es uno de los estándares para la compresión de fotografías). En este trabajo se utilizó el "BSPLAYER"¹ el cual realiza la conversión necesaria.

En esta primera etapa se trabaja con el procesador de una PC estándar (no hay procesamiento en paralelo). Este es un limitante para el procesamiento de los cuadros, pues una definición mayor a los 200.000 pixeles vuelve excesivamente lenta la evolución del programa.

Hay dos caminos a seguir:

- a) Bajamos la definición de los cuadros.
- b) Tomamos un sector de la imagen (recortamos la imagen).

Ambas soluciones tienen sus propias restricciones. En la primera, bajar demasiado la definición puede dar resultados poco precisos en el cálculo de la velocidad (bajar la definición implica descartar pixeles en la imagen). En la segunda solución, puede suceder que el sector que se recorte en cada cuadro no involucre elementos que se quieran medir produciendo en consecuencia un error en el cálculo de la velocidad.

Se opta por trabajar de los dos modos y comparar los resultados obtenidos.

Se parte de la hipótesis que la luminancia permanece constante ante movimientos espaciales en un tiempo dado. Esto permite expresarlo de la siguiente manera:

$$I(x,y,t) = I(x+dx,y+dy,t+\Delta t) \quad \text{Ecuación 1}$$

Se aplica el cálculo en serie de Taylor, donde las derivadas de orden superior pueden despreciarse sin cometer un error apreciable.

Se puede resumir las expresiones a lo siguiente

$$I_x u + I_y v + I_t t = 0 \quad \text{Ecuación 2}$$

Donde (u,v) con $u = dx/dt$ y $v = dy/dt$ corresponden al flujo óptico, I_x e I_y son las derivadas parciales vertical y horizontal de la imagen, respectivamente en el tiempo "t". Asumimos movimientos de traslación pura y el error definidos como sigue:

$$E = \sum_{x,y} \left(\frac{\partial s_c(\mathbf{x},\mathbf{y},t)}{\partial x} v_1(t) + \frac{\partial s_c(\mathbf{x},\mathbf{y},t)}{\partial y} v_2(t) + \frac{\partial s_c(\mathbf{x},\mathbf{y},t)}{\partial t} \right)^2 \quad \text{Ecuación 3}$$

Finalmente, la expresión que se va a utilizar para el cálculo de velocidades computando el error e igualando a cero, es:

$$\begin{bmatrix} \hat{v}_1 \\ \hat{v}_2 \end{bmatrix} = \begin{bmatrix} \sum_{x,y} \left(\frac{\partial I}{\partial x} \right) \left(\frac{\partial I}{\partial x} \right) & \sum_{x,y} \frac{\partial I}{\partial x} \frac{\partial I}{\partial y} \\ \sum_{x,y} \frac{\partial I}{\partial x} \frac{\partial I}{\partial y} & \sum_{x,y} \left(\frac{\partial I}{\partial y} \right) \left(\frac{\partial I}{\partial y} \right) \end{bmatrix}^{-1} \begin{bmatrix} - \sum_{x,y} \left(\frac{\partial I}{\partial x} \right) \left(\frac{\partial I}{\partial t} \right) \\ - \sum_{x,y} \left(\frac{\partial I}{\partial y} \right) \left(\frac{\partial I}{\partial t} \right) \end{bmatrix} \quad \text{Ecuación 4}$$

Esta operación se repite para cada píxel de la imagen y el resultado final una matriz de velocidades con la componente en cada eje.

¹ "www.bsplayer.com".

2.2 El Procesamiento

Una vez que se han capturado dos cuadros de video sucesivos (imágenes en formato bmp para esta primera etapa) y recordando que cada cuadro de imagen se representa como una matriz de dos dimensiones, se transforma la matriz de imagen a la norma YIQ (Luminancia representada por "Y", y la crominancia que es un plano normal a "Y" que está formada por "Q" e "I", las cuales transportan la información del color). En el caso de este estudio, se utiliza la norma YIQ pues por la hipótesis planteada, la luminancia permanece constante ante movimientos espaciales en un tiempo dado y por lo tanto se la va a involucrar en los cálculos, se determina el movimiento aparente de los patrones de desplazamiento de la intensidad.

Se define al primer cuadro como $I(\Omega, t)$, e $I(\Omega, t + \Delta t)$ para el segundo, el cual está incrementado por un intervalo de tiempo, Ω pertenece al plano (dos dimensiones).

Como primer paso se debe derivar el cuadro de imagen, se selecciona una función delta de Dirac para la tarea y se realiza el producto de convolución por su sencillez y rapidez de procesamiento. La derivación del cuadro de imagen se efectúa para los dos ejes "x" e "y", por lo tanto se realizan dos pasadas, una por las filas y otra por las columnas de la matriz imagen. Una vez lograda las dos derivadas parciales, la matriz de imagen tiene por cada pixel dos magnitudes, una representativa de la derivada en el eje "x" y otra representativa de la derivada en el eje "y". Queda como se aprecia en la figura 1.

La $I(t)$ se logra aplicando la definición de límite matemático a cada uno de los pixeles de los dos cuadros sucesivos de imagen reemplazando el Δt por el intervalo entre cuadros expresado en segundos (ejemplo si son 25 cuadros/seg, $\Delta t = 0,04$ seg.).

Ahora se tienen todos los elementos que aparecen en las ecuaciones de Lucas Kanade (ecuación 4).

Como cada pixel tiene la información de ese punto de la imagen, obtendremos el mismo número de ecuaciones como de pixeles estén involucrados.

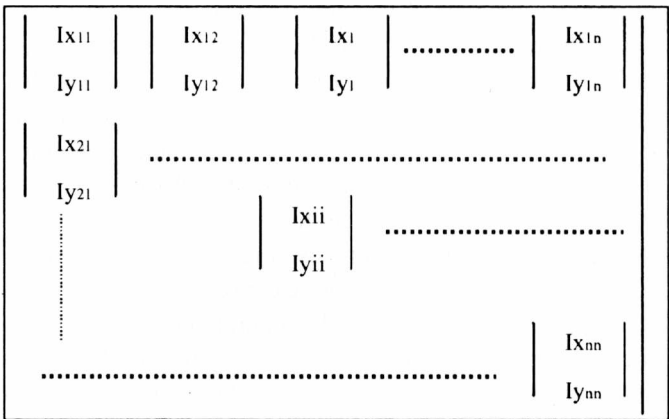


Figura 1

2.3 Resultados Obtenidos

Se procesaron distintos cuadros de imagen como podemos apreciar a continuación:

Las imágenes 1 y 2 son los correspondientes a dos cuadros de imagen sucesivos de una película. En la imagen 3 se observan pixeles que van desde un gris oscuro hasta casi un blanco en los sectores de la imagen en donde hubo un movimiento. La diferencia de luminancia (escala de grises) es la velocidad de cada uno de esos pixeles pero en forma independiente. Otra característica significativa, es que entre los dos cuadros de imagen la cámara no se movió, siendo el único movimiento el de las personas.

Nuevamente, las imágenes 4 y 5 son dos cuadros de imágenes sucesivas y el 6 es la representación de la matriz del campo de velocidades visualizada en una imagen. En este caso hay movimiento de la cámara y se pueden observar las marcas que se producen en los objetos estáticos (marcos de puertas, cuadros en la pared, etc.).

Para la verificación de los resultados arrojados por la herramienta se realizó el siguiente test:



Imagen 1

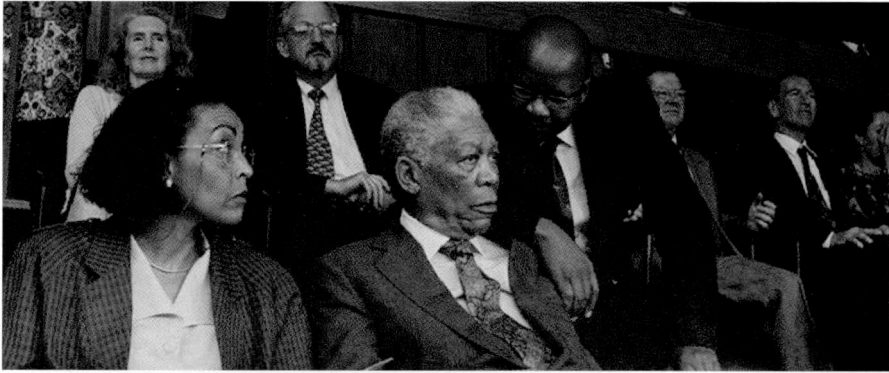


Imagen 2

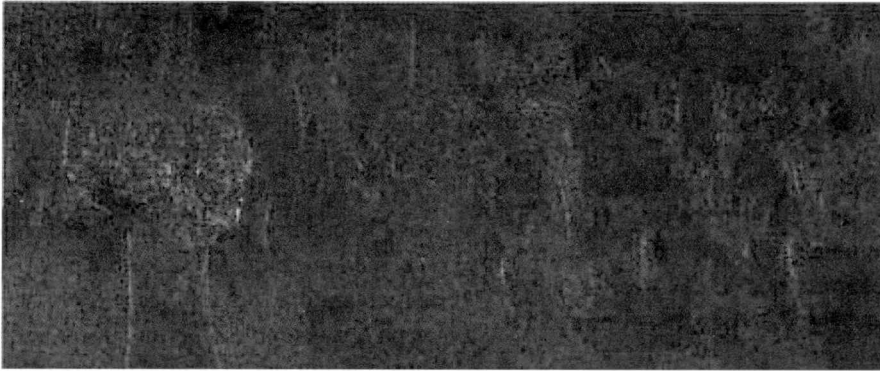


Imagen 3



Imagen 4



Imagen 5

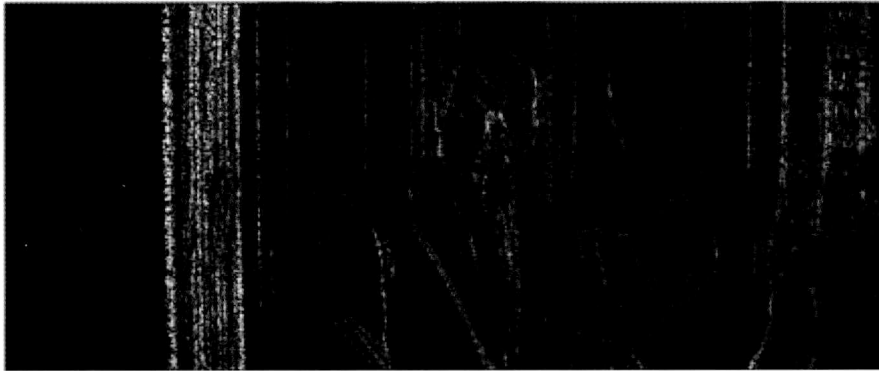


Imagen 6

2.4. Verificación de los Resultados

El cálculo que se llevó a cabo a partir de dos imágenes consecutivas y se realizó mediante el software DaVis versión 7.2, desarrollado por la empresa LaVision². Este software fue capaz a partir de las dos imágenes de devolver el campo de vectores resultado del movimiento que transforma una imagen en la siguiente.

2.5. Descripción del Software DaVis

DaVis es un completo software para aplicaciones de imágenes inteligentes (láser) donde se visualizan campos provenientes de flujos, tanto reactivos como no reactivos, así también como imágenes de superficies materiales, o el seguimiento de imágenes de alta velocidad.

La integración del software del sistema de imágenes seleccionadas se consigue mediante modos de adquisición flexible, personalizando interfaces de software DaVis y haciendo uso de paquetes de tratamiento específico de imágenes.

3. Conclusiones

El presente trabajo es el resultado de la primera etapa, se observó, a pesar de trabajar con computadoras personales de tipo notebook las cuales no permiten hacer procesamiento en paralelo, que el tiempo necesario para computar las operaciones fue menor a 1,2 segundos. Ese mismo algoritmo procesado por una GPU (placa para procesar gráficos),

la cual permite el procesamiento en paralelo, podría llegar a realizar este el mismo trabajo en 6 milisegundos. Si una cámara de video nos entrega un cuadro cada 20 milisegundos (50 cuadros por segundo trabajando en tiempo real), y sabiendo que cuando pasemos a placas graficas la velocidad aumentará más de 200 veces (para el caso de esta primera etapa sería como recibir un cuadro cada 4 segundos), podemos concluir que el tiempo de procesamiento en esta primera etapa, utilizando sistemas de procesamiento en serie, fue bueno. No obstante se sigue trabajando en la optimización del algoritmo, una de las mejoras en la que se está trabajando es partir la matriz de imagen para su procesamiento en paralelo.

De las dos opciones que se habían mencionado para bajar la definición de la imagen (bajar la definición en la cantidad de pixeles o recortar la imagen utilizando solo el sector de interés), la segunda se descarta pues los resultados obtenidos muestran que un movimiento de cámara produce el mismo efecto que si hubiera desplazamiento al recortar dos imágenes (imagen 6) y esto produce una indeterminación en la medición (no se puede determinar si el movimiento del cuadro completo se debe a movimiento de la cámara o mala selección en el recorte de imagen).

Los resultados obtenidos y corroborados con el software "DaVis" permiten vislumbrar un amplio campo de aplicaciones. Se pueden citar: estudios no invasivos del medio observado, aplicaciones directas en medicina (medición de la presión sanguínea en el endotelio) conociendo el campo de velocidades y la densidad del fluido se pueden calcular las presiones que actúan sobre las paredes contenedoras, aplicando las ecuaciones de "Navier Stokes", estudio de partículas inmersas en campos eléctricos externos, estudio del comportamiento de un grupo de insectos (se puede utilizar para seguir un objeto en particular en forma automática), etc.

² <http://www.lavision.de/en/products/davis.php>

También en aplicaciones como control automático en el flujo de fluidos, observación y corrección de vórtices y control vehicular.

Se destacan algunas particularidades:

- El algoritmo que hace la inversa de una matriz precisa ir obteniendo resultados parciales para ir obteniendo los valores de la matriz inversa, teniendo que esperar un resultado para poder hacer el próximo cálculo. Esto implica que el algoritmo no puede procesarse en paralelo. El método utilizado tiene un solo paso de inversión de matrices, y las matrices a invertir son de 2×2 , esta operación se logra con un algoritmo muy simple y de pocas instrucciones, esto da como resultado un procesamiento muy rápido frente a uno que deba invertir matrices de mayores dimensiones.
- Se deriva la función gaussiana en lugar de la función de imagen. La ventaja es la de tener una matriz muchas veces más pequeña (se puede tener una excelente resolución con una función gaussiana de 16×16 , mientras que una matriz de imagen puede tener una resolución de 1.300×1.700) y es la misma que se precisa para anali-

zar una cantidad importante de cuadros. Al ser la función gaussiana única para todo el proceso, implica que se deriva una sola vez.

- El resto del proceso (salvo la inversión de matrices), se puede paralelizar en una GPU (placa para procesar gráficos).

4. Referencias

Heitz D., Memin E., Schnörr C. (2010), "Variational Fluid Flow Measurements From Image Sequences: Synopsis And Perspectives". *Exp. Fluids*, 48(3):369-393.

Lucas B.D., Kanade T. (1981): "An Iterative Image Registration Technique With Application To Stereo Vision" Proceedings of the 7th Intl Joint Conf on Artificial Intelligence (IJCAI) 1981, August 24-28. Vancouver, British Columbia, pp.674-679.

Hoffmann G. (2001): "Gaussian Filters": <http://users.utcluj.ro/~baruch/resources/Image/gauss25092001.pdf>