

Bioinspired temporal filter modeling for motion estimation

Guillermo Botella*, José Antonio Martín H., Joaquín Recas, Concepción Sanz

*Departamento de Arquitectura de Computadores y Automática,
Universidad Complutense de Madrid. C/ José Santesmases s/n, 28040, Madrid, España.*

Tel: +34 913944392; Fax +34 91-394 7527

**gbotella@fdi.ucm.es, jamartinh@fdi.ucm.es, jrecas@fdi.ucm.es, csanzpineda @fdi.ucm.es*

Abstract

This paper models a set of temporal filters biologically plausible and their hardware implementation, being these filters the first processing stage from a robust neuromorphic motion estimation system. After designing and implementing the filters is constructed a confidence measure which modules the estimation density data using the previously implemented architecture. The global system is examined using several stimuli and error metrics showing a good behavior of the system implemented. This system has many applications regarding tracking, pattern recognition, robotics and real-time signal processing.

Keywords: artificial vision, motion estimation, robotics, bioinspired systems, digital signal processing.

Modelado de filtros temporales bioinspirados para estimación de movimiento

Resumen

Se propone en este trabajo el modelado de filtros temporales biológicamente plausibles y su implementación en hardware, siendo estos filtros la etapa inicial de un algoritmo de estimación de movimiento robusto y fuertemente bioinspirado. Posteriormente se diseña una medida de confianza que modula la densidad de cálculo de puntos a partir de los filtros temporales previamente implementados. Se examina el sistema global mediante estímulos y varias métricas de error, donde se concluye un comportamiento adecuado de los filtros y la medida de confianza construida. Este sistema tiene varias aplicaciones en reconocimiento de objetos, robótica y procesado de señal en tiempo real.

Palabras clave: visión artificial, estimación de movimiento, robótica, sistemas bioinspirados, tratamiento de señal.

1. Introducción

El tratamiento y estimación de movimiento es un campo que se relaciona con un inmenso conjunto de disciplinas y que ha cobrado una gran importancia dentro de ellas, debido a la gran cantidad de aplicaciones. Este paradigma está prácticamente en un estado incipiente; careciendo los sistemas existentes más avanzados de características que atesoran los mamíferos, además de que los recursos computacionales requere-

dos son ingentes y a menudo inabordables cuando se requiere tiempo real.

El objetivo de esta contribución es el estudio e implementación en hardware reconfigurable (bajo coste, prototipado rápido y capacidad de modificación del diseño a posteriori) de un banco de filtros temporales derivativos que forman la primera etapa de procesamiento de un modelo de flujo óptico robusto. Estos filtros serán suplementados por una modulación de los puntos a calcular.

El modelo de estimación de movimiento (McGM) usado en esta contribución, ha sido diseñado en la University College London [1], al cual se le han añadido trabajos anteriores, así como aportaciones propias [2, 3]. Se ha escogido este modelo por su robustez y fuerte bioinspiración, incorporando capacidades inherentes a los humanos, dando cuenta por ejemplo, de algunas ilusiones ópticas sólo percibidas por éstos [4, 5]. La gran contrapartida a todas estas cualidades es su complejidad, debido a las operaciones de procesado de señal realizadas, su implementación en tiempo real no es sencilla.

Respecto a la estimación de movimiento, existen tres estrategias básicas: los modelos de correlación, los de energía y los de gradiente. Los dos primeros aplican esencialmente plantillas [6, 7], siendo el objetivo de estos modelos obtener mecanismos que respondan bien al movimiento de un estímulo para una dirección y una velocidad específica, dando un ajuste entre el movimiento y la plantilla.

Obviando en este análisis preliminar otras dificultades menores, hay que señalar que la respuesta de estos dos tipos de modelos depende fuertemente del contraste del estímulo [8]. Adicionalmente, las respuestas a estas plantillas no proporcionan directamente el movimiento, teniendo que calcular la velocidad en etapas posteriores, usualmente, mediante poblaciones de filtros.

La tercera estrategia (modelos de gradiente), se basa en un cociente de derivadas temporales y espaciales en cada punto [8]. Ya que el contraste varía igualmente en el denominador y el numerador, el resultado permanece invariante al contraste sin un sobrecoste de cálculo adicional. Esta cualidad es importante, aunque deja de cumplirse en zonas donde no hay información suficiente para disparar un cierto umbral de activación. Adicionalmente al problema anterior, se añade la dificultad de la dependencia del patrón estático frente al dinámico en estas tres familias de modelos.

Consecuentemente, se hace necesario el desarrollo de otros modelos más elaborados y robustos frente a estas consideraciones, tales como modelos de gradiente (versiones previas del McGM [9]) o Lucas-Kanade [10], que solventan esta dependencia del patrón estático introdu-

ciendo una etapa de filtrado paso banda previa al procesamiento; sin embargo, este uso añade nuevas contrapartidas indeseadas:

- Al eliminar las frecuencias inferiores, se reduce la señal en patrones con un movimiento lento asociado, eliminando información potencialmente útil.
- Teniendo en cuenta el umbral inferior de movimiento en la fovea humana [5], es difícil justificar un filtro temporal biológicamente plausible que elimine la influencia del patrón estático y al mismo tiempo mantenga esta alta sensibilidad a movimientos lentos. En sistemas biológicos, se evita la influencia del patrón estático gracias a mecanismos especiales del córtex en etapas posteriores, en lugar de filtros previos. Estos modelos proporcionan una justificación para la sensibilidad al patrón estático en pacientes con ceguera al movimiento [11].

Por lo tanto, eliminar el patrón estático con el filtrado temporal no es biológicamente factible. Es necesario que la etapa de independencia al patrón estático esté ubicada al final del mismo, para que el sistema sea biológicamente plausible. Además, existen otras dificultades:

- Es necesario realizar operaciones de inversión de grandes matrices [1], que no son, de nuevo, biológicamente justificables. En este trabajo se ha adoptado una solución basada exclusivamente en respuestas de filtros orientados (mediante operaciones de suma, multiplicación y división).

El modelo usado aquí es una alternativa a sistemas previos que cumpla con los requisitos (invariancia al contraste y al patrón estático, consistencia biológica, es decir, el uso de filtros temporales y espaciales encontrados en el ser humano y el uso de operaciones realizadas mediante estos filtros). Este algoritmo atesora una serie de propiedades que no cumplen otros modelos convencionales de gradiente, como: robustez frente a ruido, consistencia matemática frente a regiones donde no hay apenas contraste, versatilidad para justificar procesamiento de movimiento de segundo orden, capacidad de separar las componentes diferenciales del flujo óptico sin recurrir a métodos iterativos, entre otras.

1.1. Introducción al modelo multicanal de gradiente

La estructura matemática y computacional del modelo multicanal de gradiente (McGM) viene descrita en [1, 9, 12], y su implementación hardware en [2, 3]. El modelo extiende la ecuación básica de conservación de la intensidad para flujo óptico, expresando la imagen como un desarrollo de Taylor e introduciendo derivadas de órdenes superiores en la descripción. Gracias a esto, se consigue una mayor robustez en la secuencia de imágenes. Mediante el cociente de la derivada temporal y espacial de varios términos del desarrollo de Taylor (calculando este desarrollo para un número de direcciones que se corresponden a las orientaciones de las columnas encontradas en el córtex visual primario), se obtiene una medida de la velocidad como función de la orientación del filtro.

Esta estrategia de múltiples canales reduce el ruido [1]. Debido a que cada cociente individual está matemáticamente mal condicionado en puntos donde el denominador es nulo, se combinan las derivadas de varios términos de este desarrollo y se integran en un volumen espacio-temporal. A partir de estos datos se genera la estimación de velocidad en módulo y fase para cada punto calculado, obteniendo finalmente un mapa denso de información.

2. Diseño de filtros apropiados para hardware

2.1. Filtrado temporal

En la práctica, la convolución temporal podría implementarse multiplicando una máscara desplazada a lo largo de la imagen de entrada y después sumar todas las contribuciones ponderadas. Hay que tener en cuenta que usando esta estructura, el número de operaciones/punto es proporcional a la extensión temporal del filtro. Los sistemas de adquisición de datos, por ejemplo captación de imágenes comerciales, almacenan en memoria posiciones correlativas de la imagen, en términos espaciales. Sin embargo, el procesado temporal opera a través de datos no adyacentes entre sí, por lo que este proceso es ineficiente y particularmente mejorable.

En la ecuación siguiente representamos el núcleo temporal del que se obtendrán derivando los operadores temporales. Esta función se construye como una Gaussiana en el espacio temporal logarítmico con los parámetros $\alpha = 10$, $\tau = 0,2$, de forma general [1, 9].

$$\text{núcleo temporal} = \frac{e^{-\left(\frac{\ln(\tau/\alpha)}{\tau^2}\right)}}{\sqrt{\pi}\alpha e^{\frac{\tau^2}{4}}} \quad (1)$$

Como se ha expuesto de forma previa, el diseñar una estructura FIR convolutiva, implica almacenar tantos fotogramas como la longitud del filtro, usando un ingente consumo de recursos que dependerá de la resolución y precisión de trabajo. A modo de ejemplo, una imagen de 128x96 puntos y 8 bits de precisión llega a consumir un 27% de la RAM externa, que corresponde a un uso del 50% de los buses de entrada/salida existentes. Por lo tanto, se necesita optimizar esta implementación mediante otro tipo de filtros menos costosos. En este trabajo se ha diseñado una forma recursiva de derivada temporal mediante la aproximación de una función más sencilla, cuya transformada z es posible calcular de forma inmediata. Se elige como entorno de desarrollo hardware Handel C [13] usando la herramienta DK [14]. Respecto a la placa de prototipado, se ha usado AlphaData RC1000 [15] que incluye el chip Virtex 2000E-BG560 [16] con 4 bancos de SRAM de 2MB.

2.2. Transformación del filtro temporal

La ecuación (2) modela la convolución de un filtro FIR, donde $x(n)$ es la señal de entrada, $y(n)$ la salida y $h(k)$ son los coeficientes del filtro, la salida tiene una realización causal:

$$y(n) = \sum_{k=0}^N h(k)x(n-k) \quad (2)$$

Por el contrario, los sistemas de respuesta impulso infinita (IIR), vienen descritos por la ecuación: (3)

$$y(n) = \sum_{k=0}^N a_k x(n-k) - \sum_{k=0}^M b_k y(n-k) \quad (3)$$

donde a_k y b_k son los coeficientes del filtro y el valor de M es el orden de recursión. Esta descripción incluye una realimentación infinita (teóricamente) de las salidas correspondientes a instantes previos, aplicando en la práctica un umbral. La ventaja de estos sistemas es que normalmente se pueden implementar con pocos coeficientes (y por consiguiente pocas multiplicaciones); su principal dificultad sin embargo, viene en la estimación de los mismos (a_k y b_k) además de su estabilidad.

La derivada de la Gaussiana aparece de forma frecuente en procesamiento de imagen, existiendo varios ejemplos de implementaciones recursivas. Vliet [17] aporta un diseño basado en una implementación recursiva, que requiere $2k$ multiplicaciones/puntos, para un orden de recursión k , siendo independiente de otros parámetros como la desviación típica σ , este método proporciona unos resultados satisfactorios. Deriche [18, 19] aplica métodos numéricos de diferenciación, para aproximar el operador LOG (Laplaciana de la Gaussiana) a una Gaussiana en el dominio espacial, siendo este método apropiado con respecto a la exactitud, aunque difícil de extrapolar a otro tipo de filtros. Jin [18] por otra parte, implementa el presente operador usando una transformada z .

2.3. Diseño de los filtros recursivos apropiados

La transformada z de la función de transferencia establece una relación entre la entrada y la salida de un sistema en el dominio de la variable compleja z . Si se aplica la transformada z a las expresiones anteriores (2 y 3), se obtiene la función de transferencia para cada tipo de filtro (5), donde $H(z)$ es la transformada de la función $h(k)$. Con respecto a la derivada de la Gaussiana $G_n(x)$, se puede realizar un desarrollo para potencias z^{-1} a partir de la expresión (4), dando lugar a (6).

$$H(x) = \sum_{k=0}^N h(k)z^{-k} \quad (4)$$

$$H(z) = \frac{a_0 + a_1Z^{-1} + a_2Z^{-2} + \dots + a_NZ^{-N}}{1 + b_1Z^{-1} + b_2Z^{-2} + \dots + b_MZ^{-N}} = \frac{\sum_{K=0}^N a_K z^{-K}}{1 + \sum_{K=1}^M b_K z^{-K}} \quad (5)$$

Llegado a este punto, es deseable relacionar los coeficientes a_k, b_k , y G_n , para ello, se realiza la división polinomial (5) para obtener una serie de potencias que puede ser comparada con (6), permitiendo resolver los coeficientes a_k y b_k . Dividiendo iterativamente, se generaliza la relación:

$$H(z) = G_n(0) + G_n(1)z^{-1} + G_n(2)z^{-2} + \dots + G_n(N)z^{-N} \quad (6)$$

$$G_k = \frac{1}{b_0} \left[a_k - \sum_{i=1}^k G_{k-1} b_i \right] \quad G_0 = \frac{a_0}{b_0}, \quad (7)$$

aunque sigue existiendo un problema de dependencia de los coeficientes G_k con los b_i . Para solventar esta dificultad se simplifica la función de transferencia IIR (6), tal y como señala Jin [18] mediante la expresión (8). En sistemas estables, el valor de p se mueve en un entorno ($0 < p < 1$), la principal ventaja de esta transformación es que el denominador es sólo función de una variable:

$$H(z) = \frac{\sum_K^N a_K z^{-K}}{(1 - pz^{-1})^N} \quad (8)$$

Se separa la derivada en dos secuencias causales ($H(z) = H^+(z) + H^-(z)$), tomando $k_s = \frac{1}{\sqrt{2ps}}$. Si se igualan (8) y la forma canónica (5), identificando potencias de z^{-1} , se pueden expresar los denominadores en función del parámetro p , determinándose su valor por ajuste iterativo. El resultado es un filtro recursivo:

$$Y^+(n) = [a_0X(n) + a_1X(n-1) + a_2X(n-2)] - [b_1Y^+(n-1) + b_2Y^+(n-2)]$$

$$Y^-(n) = [a_3X(n+1) + a_4X(n+2) + a_2X(n-2)] - [b_1Y^-(n+1) + b_2Y^-(n+2)]$$

$$Y_x(n) = Y^+(n) + Y^-(n) \quad (9)$$

donde a_k y b_k son los coeficientes del filtro. $X(n)$ es la imagen de entrada, $Y^+(n)$ es la salida intermedia de la derivada que se aplica en la dirección positiva ($Y^-(n)$ negativa). $Y_x(n)$ es la respuesta de la derivada Gaussiana, el filtro sólo requiere nueve multiplicaciones/punto, independientemente del

tamaño σ (la implementación anterior exigía 23 multiplicaciones /punto). Cada orden adicional de diferenciación requiere al menos un orden más de recursión (2 multiplicaciones más) para hacer los errores tolerables. Dichos errores se incrementan a medida que crece el orden, siendo las derivadas de órdenes elevados, computacionalmente ineficientes comparadas con la convolución directa. A continuación, se particulariza la forma de estos filtros para dotarla de una naturaleza recursiva y propia.

2.4. Forma recursiva de los filtros recursivos apropiados

En este trabajo se ha diseñado una forma recursiva de derivada temporal mediante la aproximación de una función más sencilla cuya transformada z es posible calcular de forma inmediata:

$$F_n(t) = \frac{1}{\alpha} \left(\frac{1}{\alpha}\right)^n e^{-\frac{\beta t}{\alpha}} \tag{10}$$

donde α es la escala temporal correspondiente a la Gaussiana en el dominio del logaritmo del tiempo (denominada a partir de ahora función Gelt), n y β son parámetros cuya función es optimizar el ajuste entre las dos funciones. Eligiendo cuidadosamente estos parámetros se puede imponer que ambas derivadas pasen por cero para el mismo valor de t . Para la función Gelt esto ocurre cuando $t/\alpha=1$, la función de aproximación $F_n(t)$ coincide con su pico cuando $t = \alpha n/\beta$. Por consiguiente, las dos funciones pueden hacer coincidir su pico al mismo tiempo si $\beta = n$. El valor de n toma un papel similar al de τ en la función Gelt. También se busca que $F_n(t)$ tenga su pico con el mismo valor de amplitud que la Gelt, con lo que, se llega a que esta condición se cumple en el caso siguiente:

$$\sqrt{\pi\tau} e^{\frac{\tau^2}{4}} = \left(\frac{\beta}{n}\right)^n e^n \tag{11}$$

Para cada valor de τ se puede elegir n de forma que la aproximación sea de ajuste óptimo e independiente de α . Este resultado es importante, puesto que α controla la anchura de la derivada y la posición de su pico. La ecuación (11) es trascendente y no tiene una solución analítica. La forma recursiva de $F_n(t)$ requiere que n sea entero, esta premisa limita los valores de τ para los que se pueden generar una aproximación óptima. Se puede cambiar el tamaño de la derivada sin más que cambiar α , ahorrando en etapas de diseño. La forma recursiva de $F_n(t)$ es de orden $n + 2$ y por ende, para máxima eficiencia se busca un valor bajo de n . Si la elección se hace con $n = 1$, se encuentra, como muestra la Tabla 1, que esta función se correlaciona de forma máxima con la función Gelt, para $\tau = 1, 12$, el cual tiene a su vez una anchura entre 4 y 10 fotogramas, dependiendo de α . Una comparación de Gelt y la función aproximada $F_n(t)$ puede verse en la Figura 1, mostrándose que la función $F_n(t)$ se ajusta bien a la función Gelt según α . Se va a proceder a la implementación de forma recursiva de esta función, mediante 3 procedimientos:

Se realiza la transformada z de la función FIR $F_n(t)$. Después se iguala esta expresión a la forma canónica de un filtro IIR. Por último, se obtienen a_k y b_k gracias a una comparación de potencias de z .

Se transforma la expresión anterior en una alternativa que resulte identificable con la forma canónica de los filtros IIR:

$$S_n = \sum_{k=1}^n kx^k \text{ y } x = z^{-1}e^{-\frac{1}{\alpha}} \tag{12}$$

Combinando estas expresiones mediante multiplicaciones y restas, obtenemos la expresión siguiente:

$$S_\infty = \frac{x}{(1-x)^2} \tag{13}$$

Tabla 1
Solución de la ecuación (11) para diferentes valores de n y τ

n	1	2	3	4	5	6
τ	1,12	1,82	2,46	3,04	3,55	5,55

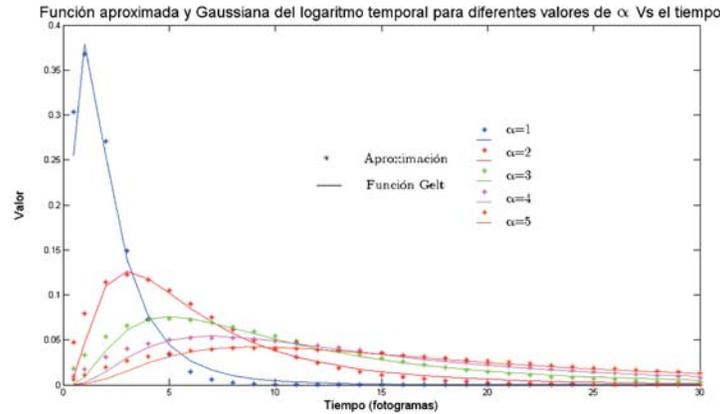


Figura 1. Diferentes funciones Gaussianas en el logaritmo temporal y sus funciones aproximadas. Nótese como la aproximación es satisfactoria independientemente del valor de α .

Sustituyendo $x = z^{-1}e^{-\frac{1}{\alpha}}$, tenemos la expresión general de la transformada z de la función aproximada:

$$H(z) = \frac{z^{-1}e^{-\frac{1}{\alpha}}}{(1 - z^{-1}e^{-\frac{1}{\alpha}})^2}. \quad (14)$$

Llamando $\gamma = 1/\alpha$ e igualando a la expresión general de la forma canónica de un filtro IIR:

$$H(z) = \frac{\sum_{k=0}^N a_k z^{-k}}{1 + \sum_{k=1}^M b_k z^{-k}} = \frac{1}{\alpha^2} \frac{ze^{-\gamma}}{(z - e^{-\gamma})^2}. \quad (15)$$

Es posible determinar los coeficientes b_k , a_k por comparación con la ecuación (9), deduciéndose:

$$a_0 = 0; a_1 = \frac{1}{\alpha^2} e^{-\gamma}; b_1 = -2e^{-2\gamma}; b_2 = e^{-2\gamma}. \quad (16)$$

Aplicando la transformada inversa, se llega a la implementación recursiva temporal deseada:

$$y(n) = \frac{1}{\alpha^2} x(n-1)e^{-\gamma} + 2e^{-\gamma}y(n-1) - e^{-2\gamma}y(n-2) \quad (17)$$

Esta expresión denota que la función aproximada puede generarse a partir de la imagen de entrada y 2 fotogramas previos de salida. Se necesitan 3 multiplicaciones y 2 sumas; si se compara con la convolución temporal usual, ésta requiere el mismo número de multiplicaciones y

sumas que el tamaño por defecto de la ventana temporal (23 fotogramas). Incrementamos pues el rendimiento en un factor 7.

2.5. Implementación hardware de los filtros temporales diseñados al efecto

A continuación se procede a implementar físicamente esta estructura, según un triple buffer circular. Este proceso se vuelve a repetir cada 3 fotogramas consecutivos, mostrándose la existencia de entradas para cada uno de los pesos ponderados que intervienen en el proceso de cálculo. Posteriormente, es necesario implementar un módulo que se encargue de formar las derivadas temporales, debido a que el diseñado anteriormente sólo crea la derivada de orden cero; para aprovechar recursos de la estructura presente diseñada, se idea una derivación numérica basada en simples operadores de gradiente, con una longitud asociada de dos y tres operaciones respectivamente [3].

Iterando sobre el operador gradiente de primer orden, se obtiene el segundo orden. En las estructuras denominadas T0, T1 y T2 se van almacenando los resultados de la derivada de orden cero, primero y segundo. Es posible implementar la derivada con operadores más complejos, evitando introducir errores acumulativos según se van efectuando las iteraciones para conformar órdenes superiores, además de que a medida que las frecuencias crecen, sus errores se incrementan. Usando sinusoides y para una longitud constante, los operadores tienen errores

asociados que resultan inversamente proporcionales a la frecuencia fundamental del armónico [20].

Adicionalmente, otro aspecto a destacar es el efecto del filtro paso baja, considerando que inhibe el error de los operadores discretos en el cálculo del flujo óptico cuando se usan modelos de gradiente [21]. Después de modelar el filtro, se implementa con diferente profundidad de bits. En la Figura 2 mostramos la respuesta en frecuencia y respuesta impulso para implementaciones de 4, 8 y 16 bits (aritmética en punto fijo). Se puede ver que con 16 bits de precisión obtenemos unos resultados con un error inapreciable.

3. Métricas de error y filtrado de puntos

3.1. Métricas de error usadas para los experimentos

A continuación vemos someramente las métricas más usadas de error de flujo óptico con objeto de evaluar los filtros diseñados en el contexto de los métodos de estimación de movimiento. Las medidas más sencillas son los errores absolutos de módulos y fases:

$$\text{Error}_{\text{FASE}} = \left| \arctan\left(\frac{\vec{v}_{ey}}{\vec{v}_{cx}}\right) - \arctan\left(\frac{\vec{v}_{ey}}{\vec{v}_{ex}}\right) \right|;$$

$$\text{Error}_{\text{MODULO}} = \left| \|\vec{v}_e\| - \|\vec{v}_c\| \right|, \tag{18}$$

donde v_c es la velocidad correcta y v_e la velocidad estimada. La medida más usada para estimar flujo óptico es propuesta por Barron [22] con el error angular entre la velocidad correcta v_c y la estimada v_e .

$$\vec{v} = \frac{(v_x, v_y, 1)}{\sqrt{v_x^2 + v_y^2 + 1}}; \text{Error}_{\text{BARRON}} = \arccos(\vec{v}_c, \vec{v}_e) \tag{19}$$

Siendo esta métrica aceptada, existen alternativas [23] reflejadas en las expresiones (20), donde aparece la diferencia entre v_c y v_e , y una medida del error normal a la dirección del gradiente:

$$\text{Error}_{\text{GALVIN}} = \|\vec{v}_c - \vec{v}_e\|;$$

$$\text{Error}_{\text{GALVIN}\perp} = \|(\vec{v}_c - \vec{v}_e) \cdot \hat{g}^\perp\|. \tag{20}$$

Como punto de partida se muestra el error evaluado según las métricas anteriores. En la Figura 3, se presenta una muestra de la salida correspondiente a las 3 derivadas y en la Tabla 2 se muestra el error para diferentes métricas. Mostramos los estímulos en la Figura 4.

3.2. Modelado de la medida de confianza

Cuando existen variaciones temporales muy pequeñas, asociadas a la intensidad, existe una causa de error añadido correspondiente, por un lado, a las fuentes de ruido y, por otro, a movimientos muy lentos. Se modela una medida de

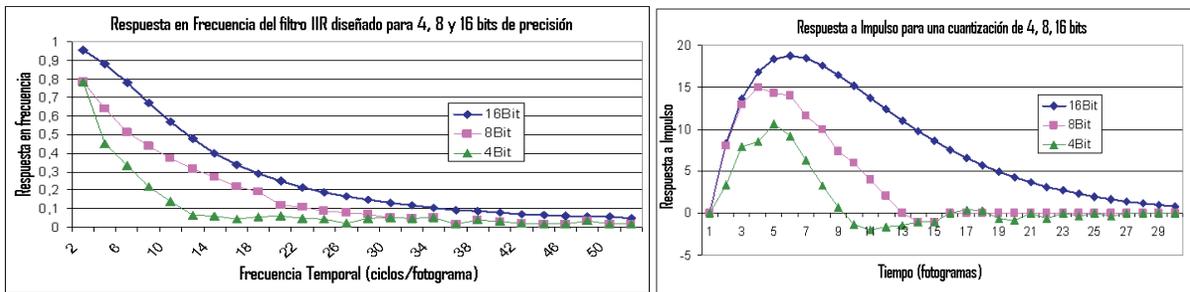


Figura 2. (Izquierda) Respuesta en frecuencia para la implementación del filtro IIR paso baja con 4, 8, y 16 bits respectivamente. (Derecha) Respuesta impulsiva de esta implementación.

Tabla 2. Error absoluto del sistema global según las diferentes métricas

Módulo	Fase	Barron	Galvin	Galvin ⊥
0,04	1,6	1,55	0,02	0,02

confianza como un selector de puntos de cálculo, definida por la expresión:

$$\left| \frac{dI}{dt} \right| + \left| \frac{d^2I}{dt^2} \right| \leq \text{UMBRAL_MASCARA} \quad (21)$$

donde I hace referencia a la intensidad de entrada de la imagen. Esta medida de confianza (21), filtrará los puntos cuyas variaciones temporales asociadas no sean superiores a un umbral definido; Barron [22] sugiere algo similar. Se va proceder al ensayo con un valor de máscara de 0,1 y como primera y segunda derivadas se van a usar los filtros temporales diseñados en este trabajo. Como resultado, se comprueba experimental-

mente una mejora de los errores respecto a este umbral, como contrapartida, se constata una pérdida de puntos debido al filtrado (que no debe confundirse con la pérdida de puntos debida a la precisión), tal y como se ilustra en la Figura 5 (izquierda). La medida de confianza diseñada es apropiada, disminuyendo el error para todas las métricas. Por último, en la Figura 5 (derecha) mostramos los resultados del sistema completo [3, 4] donde se emplean los filtros recursivos.

4. Resultados

Se presentan los resultados del modelo mejorado, respecto a la implementación de los tra-



Figura 3. Resultados de los filtros recursivos a una secuencia de adelantamiento de vehículos. De izquierda a derecha, derivadas de orden 0, 1, 2. Secuencia de entrada proporcionada por HELLA [24].

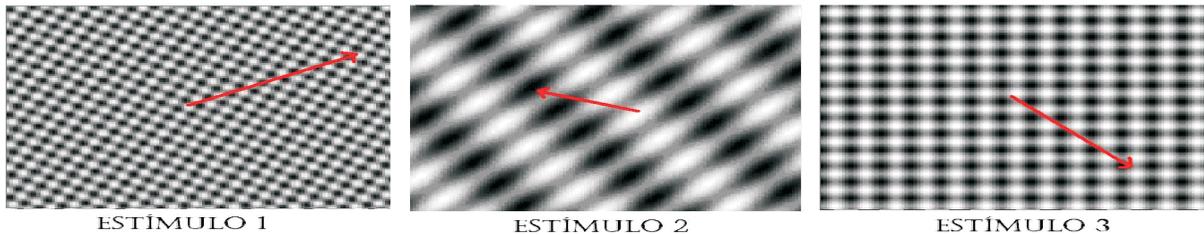


Figura 4. Ilustración de tres estímulos para pruebas: patrón moviéndose a 2.05, 1.0, y 1.4142 puntos/fotogramas (módulo) y 28°, 162° y -45° (orientación) de izquierda a derecha.

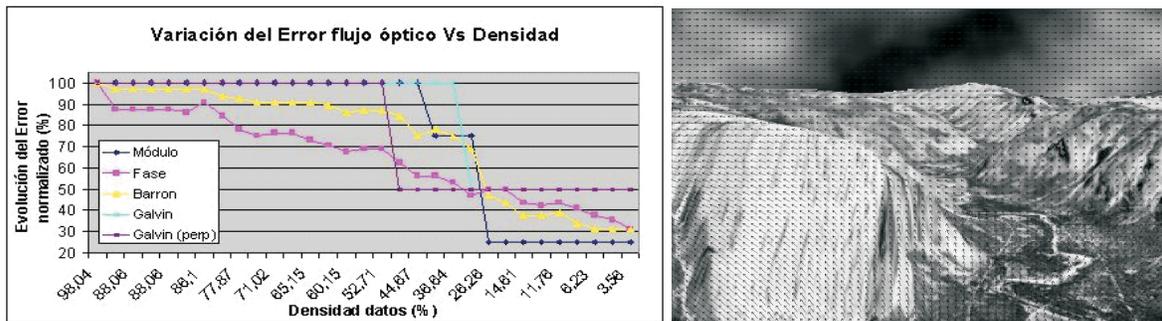


Figura 5. (Izquierda) Evolución del Error (normalizado con el máximo de Tabla 2) frente a la densidad al aplicar los filtros. (Derecha) Salida de la secuencia de Yosemite, (<http://www.cs.brown.edu/~black/images.html>). Errores inferiores a 4° según métrica de Barron [22].

Tabla 3
Rendimiento de la implementación actual frente a las presentadas [2, 3].
Resultados en Mega Puntos por segundo (Mpps) y Fotogramas por segundo (Fps)
para una resolución 150×150 puntos

Modelo Previo Mpps	Modelo Previo (Fps)	M. Actual (Mpps)	M. Actual (Fps)
5,1	226	7,6	337

bajos previos [2, 3]. La última versión implementada se basa en un prototipo refinado con posibilidad de definir la precisión exacta de las derivadas temporales (etapa I), donde sólo se usaban operadores numéricos $[-1, 1]$. Las derivadas espaciales del núcleo inicial (etapa II), son realizadas con derivadas numéricas de diferentes órdenes de forma recursiva y con un control de la precisión absoluto, sin embargo en trabajos anteriores [2, 3] se realiza una convolución separable y cuantizada, pero no de forma numérica, es decir con una longitud de ventana espacial igual al filtro a operar. Otra novedad es que se permiten filtros espaciales de forma de que se use una estructura piramidal hasta 9° orden, no habiendo ninguna versión hasta ahora con esta longitud. (45 filtros derivativos/pirámide). La etapa de orientaciones (etapa III) es capaz de sintetizar filtros orientados hasta 90 ángulos (cada 4 grados). El resto de etapas han sido optimizadas de forma que se ha implementado el modelo con un paradigma multi-escala, adaptándose su precisión a las circunstancias del entorno. En la Tabla 3 se muestra el rendimiento de la presente implementación

El modelo mejora hasta aproximadamente 1,5× el modelo publicado en [2, 3]. Respecto al comportamiento de la medida de confianza para todas las métricas analizadas, este filtro hace que la diferencia de módulos, métrica de Galvin y Galvin perpendicular permanezca invariante hasta computar el flujo óptico de aproximadamente un 40%, 30% y 45% de los puntos de entradas respectivamente. La métrica de Barron y la Fase tienen un comportamiento gradual decreciente desde un 90% hasta un 3%.

5. Conclusión

Se ilustra una técnica para diseño e implementación de filtros temporales recursivos apropiados para estimación de movimiento y visión

artificial. El sistema se ha implementado en hardware reconfigurable tipo FPGA. Hemos complementado este trabajo aplicando una medida de confianza que filtra la densidad de puntos a calcular según los filtros temporales diseñados. Por último, se ha comprobado la bondad de la medida confianza diseñada que filtra puntos en la ruta de datos, donde este último mecanismo modula la calidad del resultado final en función del número de puntos cuyo flujo debe ser estimado. Como conclusión, se ha verificado una disminución del error respecto a la densidad de puntos en la ruta de datos, independiente de la métrica de error considerada, relacionando la calidad con la densidad de puntos de procesamiento. Este es uno de los pocos trabajos que nos conste que se examinen resultados de estimación de movimiento evaluados por cinco métricas diferentes. Estos sistemas tienen gran utilidad en seguimiento de objetos, robótica y cibernética.

Referencias bibliográficas

1. Johnston A., McOwen P. W., Benton C.: "Robust Velocity Computation from a Biologically Motivated Model of Motion Perception". Proc. Royal Society of London B Vol. 266 (1999) 509-518.
2. Botella, G., Meyer-Baese, U., García. A.: "Bioinspired Robust Optical Flow Processor System for VLSI Implementation". IEE Electronic Letters), Vol. 45 No 25, (2009) 1304-1306.
3. Botella, G., García. A., Rodríguez. M, Ros, E., Meyer-Baese, U., Molina M.C.: "Robust Bioinspired Architecture for Optical Flow Computation". IEEE Transactions on Very Large Scale Integration (VLSI) System. Vol. 18 No 4, (2010) 616-629.
4. Anderson, A., McOwan, P.: "Humans deceived by predatory stealth strategy camou-

- flaging motion". Proceedings of the Royal Society, Vol. 720, (2003) 18-20.
5. Johnston A., Clifford C. W. G.: "A Unified Account of Three Apparent Motion Illusions", Vision Research Vol. 35 No 8, (1994) 1109-1123.
 6. Huang C., Chen Y.: "Motion Estimation Method Using a 3D Steerable Filter". Image and Vision Computing, Vol. 13, (1995) 21-32.
 7. Oh H., Lee H.: "Block Matching algorithm based on a adaptive reduction of the search area for motion estimation". Real Time Imaging, Vol. 6, (2000) 407-414.
 8. Johnston A. y Clifford C. W. G: "Perceived Motion of Contrast modulated Gratings: Prediction of the McGM and the role of Full-Wave rectification". Vision Research Vol. 35, (1995) 1771-1783.
 9. Johnston, A., McOwan, P.W. and Buxton, H.: "A computational model of the analysis of first and second order motion by simple and complex cells." Proc. R. Soc. Lond. B., Vol. 250, (1992) 297-306.
 10. Baker, S., Matthews, I. "Lucas-Kanade 20 Years On: A Unifying Framework". Int. Journal of Comp.Vision, Vol. 56, No 3, (2004) 221-255.
 11. Hess, R. H., Baker, C. L. & Zihl, J.: "The motion-blind patient: low-level spatial and temporal Filters". J. Neuroscience, Vol. 9, (1989) 1628-1640.
 12. Johnston, A., McOwan, P.W. y Benton, C.P.: "Biological computation of image motion from flows over boundaries". J Physiol. Paris, Vol. 97, (2003) 325-34.
 13. "Handel-C Lenguaje reference manual and DK tool". Celoxica company, (2007).
 14. [on-line] <http://www.mentor.com/products/fpga/handel-c/dk-design-suite/>
 15. [on-line] (AlphaData RC1000 product) <http://www.alpha-data.com/adc-rc1000.html>, (2006).
 16. Xilinx company (2011): http://www.xilinx.com/support/documentation/data_sheets/ds022-1.pdf
 17. Van Vliet L. J., Young I.T, Verbeek P.M.: "Recursive Gaussian Derivative Filters". Proceedings of the XIV International Conference on Pattern Recognition (1998) 509-514.
 18. Jin J. S. Gao Y. "Recursive Implementation of LoG Filtering". Real-Time Imaging 3, (1997) 59-65.
 19. Burden R., Douglas J.: "Análisis Numérico". ISBN 9-7068-6608-6 Thomson Paraninfo (2007).
 20. Spiegel M.: "Calculus of Finite Differences and Differential equations". McGraw Hill. (1971).
 21. Christmas W. J.: "Spatial Filtering Requirements for Gradient- Base Optical Flow Measurements". Proceedings of the British Machine Vision Conference, (1998)185-194.
 22. Barron J. L., Fleet D. J., and Beauchemin S. S. "Performance of optical flow techniques". International Journal of Computer Vision, Vol. 12, No. 1, (1994) 43-77.
 23. Galvin B., McCane B., Novins K., Mason D. & Mills S.: "Recovering Motion Fields: An Evaluation of Eight Optical Flow Algorithms". Proc. of BMVC '98. (1998)
 24. [online]: <http://www.hella.com/produktion/HellaES/WebSite/Channels/Home/Home.jsp> (2007).

Recibido el 23 de Marzo de 2012
En forma revisada el 28 de Enero de 2013