

Transformada multiescala de Radon: más allá de las 2D y el particionado binario

Fernando Rosa, José G. Marichal-Hernández, José M. Rodríguez-Ramos

E-mail {jmarih, frosa, jmramos}@ull.es

Dpto. de Física Fundamental y Experimental, Electrónica y Sistemas.

Universidad de La Laguna. Avda. Francisco Sánchez, s/n, 38205, San Cristóbal de La Laguna.

Resumen—This work revisits and deepens on the multiscale approach proposed in the last decade by Götz and Druckmüller, and simultaneously by Brady, to compute the approximate Radon transform of discrete two dimensional data. This approach is specially relevant due to its low computational demands, as long as the field of application admits its equally lower quality.

We will show how to extend the method to tackle more than two dimensions simultaneously, and how to work on non power of two domains.

I. INTRODUCCIÓN

La transformada analítica de Radon, $\mathfrak{R}f(\theta, \rho)$, de una función $f(x, y)$, con $(x, y) \in \mathbb{R}^2$, calcula las integrales de dicha función sobre el espacio de líneas parametrizadas por θ , ángulo de las líneas respecto del eje y , y ρ , distancia al origen [1]. Esto es,

$$\mathfrak{R}f(\theta, \rho) = \iint f(x, y) \delta(x \cos \theta + y \sin \theta - \rho) dx dy, \quad (1)$$

o, alternativamente, usando la forma «pendiente-desplazamiento», «slope-displacement», con $|s| < 1$,

$$\mathfrak{R}f(s, d) = \int f(u, u s + d) du, \quad (2)$$

para las regiones con $|\theta| \leq \frac{\pi}{2}$, o lo que es lo mismo, $|y| \leq |x|$, y $\mathfrak{R}f(s, d) = \int f(u s + d, u) du$, para $|\theta| \geq \frac{\pi}{2}$.

Cuando se trata de dominios discretos, hablaríamos de la transformada discreta de Radon, *DRT*. Para computarla existen algoritmos de complejidad cuasi-lineal. Es el caso de los denominados métodos multiescala, en los cuales, las integrales de línea se aproximan por sumatorios de los valores en los nodos —píxeles si hablamos de una imagen— ubicados en la correspondiente banda de ancho unitario, como proponen Brady, en [2] y [3]; Götz & Druckmüller en [4]; y Brandt & Dym en [5]; e ilustramos en la figura 2.

II. MODIFICACIONES A LA 2D ADRT DE G&D

Consideremos la definición de la *DRT aproximada* dada por G&D, para una imagen de tamaño $N \times N$, con $N = 2^n$, $n \in \mathbb{Z}$.

$$\tilde{f}(s, d) = \sum_{\mathbf{u} \in \mathbb{Z}_2^n} f(\lambda(\mathbf{u}), l_s^n(\mathbf{u}) + d), \quad (3)$$

donde $\lambda(u_0, \dots, u_{n-1}) = \sum_{i=0}^{n-1} 2^i u_i$ traduce un vector de coeficientes binarios $\mathbf{u} = (u_0, \dots, u_{n-1}) \in \mathbb{Z}_2^n$, a su correspondiente entero, $u \in \mathbb{Z}$; y la función $l_s^n : \mathbb{Z}_2^n \rightarrow \mathbb{Z}$, definida

como,

$$l_s^0(\mathbf{u}) = 0, \\ l_s^n(u_0, \dots, u_{n-1}) = l_{\lfloor s/2 \rfloor}^{n-1}(u_0, \dots, u_{n-2}) + u_{n-1} \left\lfloor \frac{s+1}{2} \right\rfloor$$

es la que determina la discretización de la línea recta $y = x \frac{s}{N-1} + d$, que aproximamos con la banda de un píxel de ancho que atraviesa la imagen a través del conjunto de píxeles con coordenadas $\{(0, d), \dots, (\lambda(\mathbf{u}), l_s^n(\mathbf{u}) + d), \dots, (N-1, s+d)\}$.

El sumatorio en \mathbf{u} se puede descomponer en múltiples sumatorios en índices binarios u_0, \dots, u_{n-1} , los cuales se pueden resolver de uno en uno, obteniendo el resultado deseado tras n etapas de transformación parcial:

$$\begin{array}{ccc} f(x, y) & \xrightarrow{\text{etapas}} & \tilde{f}(s, d) \\ \Downarrow & & \Downarrow \\ \tilde{f}^0(\mathbf{v}, d) & \xrightarrow{\text{etapas}} \tilde{f}^m(\mathbf{v}, d) \xrightarrow{\text{etapas}} & \tilde{f}^n(\mathbf{v}, d) \end{array}$$

En las etapas intermedias, considerando m el número de índices ya transformados, la *aDRT* parcial resulta

$$\tilde{f}^m(\mathbf{v} | d) = \tilde{f}^m(\underbrace{v_0, \dots, v_{m-1}}_{\underline{\mathbf{v}}}, \underbrace{v_m, \dots, v_{n-1}}_{\bar{\mathbf{v}}} | d) = \sum_{\mathbf{u} \in \mathbb{Z}_2^n} f(\lambda(\mathbf{u}, \bar{\mathbf{v}}) | l_{\lambda(\text{bit-reversal}(\underline{\mathbf{v}}))}^m(\mathbf{u}) + d). \quad (5)$$

Donde hemos usado el caracter $|$ para separar los parámetros en f , como en $f(x | y)$; y reservamos las comas para separar las dimensiones binarias en las que descomponemos a su vez cada parámetro, como en $f(x_0, x_1 | y_0, y_1)$.

II-A. Separando la pendiente de las dimensiones espaciales

Proponemos dos modificaciones a la notación de G&D.

Considérense los roles de los m índices binarios menos significativos de \mathbf{v} , denotados por $\underline{\mathbf{v}}$, y los $n-m$ Más Significativos, $\bar{\mathbf{v}}$. Es conveniente distinguir entre ellos, en lugar de considerar que existe un \mathbf{v} que los engloba, dado que este \mathbf{v} global no es un parámetro que podamos definir geoméricamente de manera natural, sino la mezcla de dos que sí lo son: la primera parte, $\underline{\mathbf{v}}$, contiene la información relativa a los bits ya transformados de la pendiente; mientras que $\bar{\mathbf{v}}$ representa

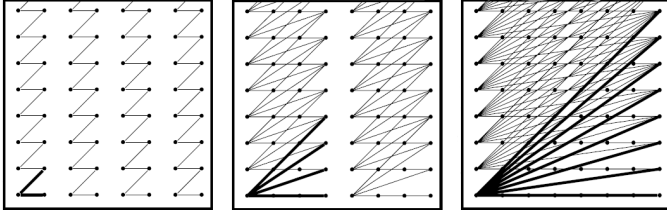


Figura 1. Combinación multiscala de segmentos de línea.

a los aún no transformados en la dimensión espacial, en este caso x , sobre la que se lleva a cabo la integración¹. Por lo que proponemos trabajar con \tilde{f}^m compuesto por tres parámetros: la *pendiente*, la *franja horizontal* y el *desplazamiento vertical*.

Nótese también que el transformación de índices desde la variable de integración a la pendiente, se da entre posiciones menos significativas en un lado y Más Significativas en el otro. Por lo que, si se reubican en las transformadas parciales convenientemente según van apareciendo, nos podemos ahorrar la inversión final de bits que describe [4].

En resumen, los cambios que introduciremos con respecto a la notación usada en (5) son: *a)* denotar la pendiente, antes \underline{v} , por s ; *b)* distinguir entre el índice binario de la pendiente que se transformará en la etapa m , s_{n-m} , de los ya transformados, σ ; y *c)* denotar los índices Más Significativos de la dimensión de integración, anteriormente \bar{v} , como nuestra \mathbf{v} .

La fórmula resultante es

$$\tilde{f}^m(\overbrace{s_{n-m}, s_{n-m+1}, \dots, s_{n-1}}^s \mid \overbrace{v_m, \dots, v_{n-1}}^{\mathbf{v}} \mid d) = \sum_{\mathbf{u} \in \mathbb{Z}_2^m} f(\lambda(\mathbf{u}, \mathbf{v}) \mid l_{\lambda(s)}^m(\mathbf{u}) + d). \quad (6)$$

Esta parametrización de la aDRT 2D parcial, \tilde{f}^m , conduce a la siguiente ecuación de mapeo entre dos etapas consecutivas,

$$\tilde{f}^{m+1}(\overbrace{s_{n-m-1}, \overbrace{s_{n-m}, \dots, s_{n-1}}^{\sigma: m \text{ bits}}}^{s: m+1 \text{ bits}} \mid \overbrace{v_{m+1}, \dots, v_{n-1}}^{\mathbf{v}: n-m-1 \text{ bits}} \mid d) = \tilde{f}^m(\sigma \mid 0, \mathbf{v} \mid d) + \tilde{f}^m(\sigma \mid 1, \mathbf{v} \mid d + s_{n-m-1} + \lambda(\sigma)) \quad (7)$$

según se detalla en [4, sección IV].

III. EXTENSIÓN A 3 Y 4 DIMENSIONES

Antes de introducir las aDRT para 3D y 4D, analicemos sucintamente el funcionamiento del método multiscala para el caso 2D. Para mayores detalles referirse a [3].

En la figura 1, tomada de ese trabajo, observamos como es posible evaluar todas las líneas² que atraviesan el plano

¹En el texto nos referiremos por variable de integración a la que ejerce de índice del sumatorio discreto.

²El método que se describe cubre el rango de pendientes $[0^\circ: 45^\circ]$. Para considerar los 180° , habría que repetir el proceso otras 3 veces modificando adecuadamente la disposición de los datos o el algoritmo, para que operase sobre ese rango alterado en $+45^\circ$, $+90^\circ$ y $+135^\circ$.

si reducimos inicialmente el problema hasta llevarlo a su expresión mínima, y procedemos luego, poco a poco, a doblar su tamaño. La etapa más básica es aquella que considera el plano dividido en franjas de ancho 2 a lo largo de la variable de integración.

A esa escala, toda «línea discreta» —según la discretización fijada mediante la ec. (4)— que forma parte del haz que cruza por un cierto nodo, tiene solo dos posibilidades a su paso por éste. Incrementar su pendiente (piénsese que más que pendiente s representa un *ascenso*), o mantenerla. Esto es, localmente, toda línea que atraviesa el nodo $(2x, y)$ incluye a uno de los dos segmentos de línea que ligan dicho punto con $(2x+1, y)$ y $(2x+1, y+1)$. Estos segmentos son las líneas más simples posibles —no hay segmentos de menos de 2 nodos—, pero contienen ya toda la información necesaria para resolver el problema a mayor escala. No hay combinación de nodos vecinos a escala 2 que no haya sido tenida en cuenta, por lo que es solo cuestión de operar adecuadamente reusando estos segmentos, para dar lugar a todas las combinaciones de «líneas discretas» a través de 4 nodos. Luego, con las de 4 nodos montar las de 8, y así sucesivamente.

Esta idea se plasma formalmente mediante las cuatro ecuaciones que hemos visto para el caso 2D, y son las que podemos extender, por ejemplo a 3 y 4 dimensiones:

- La definición analítica, $\mathfrak{R}f(s, d)$, ec. (2). Que determina cuál será la transformada integral en el continuo. Sobre el plano, la transformada de Radon, evalúa integrales de línea. En un dominio de mayor dimensionalidad se abre un abanico de posibilidades acerca de cuántas y cuáles dimensiones integrar, y cómo se relacionan las variables de integración con las no integradas. Por lo que será importante comenzar definiendo el problema continuo que aproximaremos.
- La transformada aproximada, $\tilde{f}(s, d)$, ec. (3); y la transformada parcial, \tilde{f}^m , ec. (6). Donde un sumatorio sustituye a la integral, y donde las relaciones entre las variables de integración y las no integradas se rigen por la función (4). Conllevan la división implícita del problema en «múltiples escalas». Los resultados intermedios, los mismos que ilustra la figura 1, no son otra cosa que la evaluación parcial de los sumatorios. Así, en el caso 2D, \tilde{f}^1 contiene las sumas sobre los segmentos de tamaño 2, \tilde{f}^2 sobre los de ancho 4, etcétera.
- El mapeo entre etapas consecutivas, $\tilde{f}^m \rightarrow \tilde{f}^{m+1}$, ec. (7). Las ecuaciones anteriores finalmente conducen a ésta, que es la clave del algoritmo, y contiene el modo en que combinar los resultados en cada etapa para dar lugar a los siguientes en el escalafón. Por ejemplo, en la figura 2 se ilustra cuáles son los datos involucrados en el cómputo de un único elemento, $\tilde{f}^3(4, 1)$, sobre una rejilla de tamaño 8×8 : qué segmentos de tamaño 4, y, antes, cuáles de tamaño 2, contribuyen a la línea de pendiente 4 que parte de la posición vertical 1. Volviendo sobre los resultados intermedios que quedan

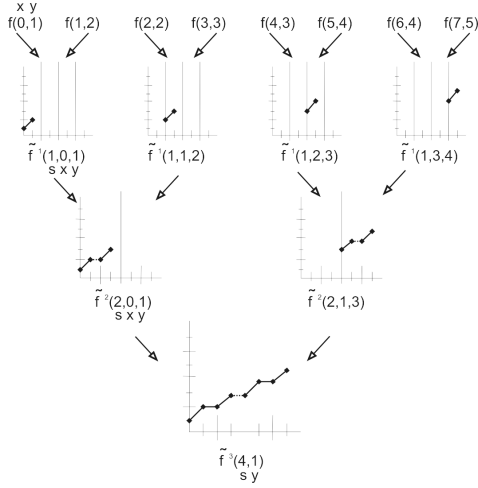


Figura 2. Sumas involucradas en el cómputo del valor $\tilde{f}^3(4,1)$ sobre una rejilla 8×8 .

definidos en las transformadas parciales, nótese que el número de franjas horizontales se reduce a la mitad a cada incremento de m , pero cada una es el doble de ancha. Igualmente se dobla el máximo ascenso que puede darse en vertical. Esto se debe a la transferencia de índices binarios de una a otra dimensión. Inicialmente, no existirán índices en la pendiente, aparecerán gradualmente, y finalmente, cuando $m=n$, solo quedará pendiente, mientras que la información horizontal habrá desaparecido.

Con esto en mente, procedemos a definir escuetamente las ecuaciones que rigen las transformadas aproximadas discretas de Radon en 3D y 4D.

III-A. Transformada aproximada de Radon en 3D

Las ecuaciones que evalúan (aproximadamente) las integrales a lo largo de los planos descritos por $z - x \text{ slope}_x - y \text{ slope}_y - \text{offset} = 0$, en un volumen $f(x, y, z)$, son las siguientes.

En el continuo, definimos la transformada integral

$$\mathfrak{R}f(s_1, s_2, d) = \int f(u_1, u_2, u_1 s_1 + u_2 s_2 + d) du_1 du_2, \quad (8)$$

cuya aproximación discreta es

$$\tilde{f}(s_1 | s_2 | d) = \sum_{\mathbf{u}_1 \in \mathbb{Z}_2^n} \sum_{\mathbf{u}_2 \in \mathbb{Z}_2^n} f(\lambda(\mathbf{u}_1) | \lambda(\mathbf{u}_2) | l_{s_1}^n(\mathbf{u}_1) + l_{s_2}^n(\mathbf{u}_2) + d). \quad (9)$$

Si definimos la solución hasta la etapa m como

$$\tilde{f}^m(\overbrace{s_{1n-m}, \dots, s_{1n-1}}^{\mathbf{s}_1} | \overbrace{v_{1m}, \dots, v_{1n-1}}^{\mathbf{v}_1} | \overbrace{s_{2n-m}, \dots, s_{2n-1}}^{\mathbf{s}_2} | \overbrace{v_{2m}, \dots, v_{2n-1}}^{\mathbf{v}_2} | d) = \sum_{\mathbf{u}_1 \in \mathbb{Z}_2^m} \sum_{\mathbf{u}_2 \in \mathbb{Z}_2^m} f(\lambda(\mathbf{u}_1, \mathbf{v}_1) | \lambda(\mathbf{u}_2, \mathbf{v}_2) | l_{\lambda(\mathbf{s}_1)}^m(\mathbf{u}_1) + l_{\lambda(\mathbf{s}_2)}^m(\mathbf{u}_2) + d), \quad (10)$$

la solución se obtiene aplicando recursivamente el siguiente mapeo entre etapas.

$$\begin{aligned} & \tilde{f}^{m+1}(\overbrace{s_{1n-m-1}, \dots, s_{1n-1}}^{\mathbf{s}_1} | \overbrace{v_{1m+1}, \dots, v_{1n-1}}^{\mathbf{v}_1} | \overbrace{s_{2n-m-1}, \dots, s_{2n-1}}^{\mathbf{s}_2} | \overbrace{v_{2m+1}, \dots, v_{2n-1}}^{\mathbf{v}_2} | d) = \\ & \tilde{f}^m(\mathbf{s}_1 | 0, \mathbf{v}_1 | \mathbf{s}_2 | 0, \mathbf{v}_2 | d) + \\ & \tilde{f}^m(\mathbf{s}_1 | 1, \mathbf{v}_1 | \mathbf{s}_2 | 0, \mathbf{v}_2 | d + s_{1n-m-1} + \lambda(\mathbf{s}_1)) + \\ & \tilde{f}^m(\mathbf{s}_1 | 0, \mathbf{v}_1 | \mathbf{s}_2 | 1, \mathbf{v}_2 | d + s_{2n-m-1} + \lambda(\mathbf{s}_2)) + \\ & \tilde{f}^m(\mathbf{s}_1 | 1, \mathbf{v}_1 | \mathbf{s}_2 | 1, \mathbf{v}_2 | d + s_{1n-m-1} + \lambda(\mathbf{s}_1) + s_{2n-m-1} + \lambda(\mathbf{s}_2)) \end{aligned} \quad (11)$$

III-B. Transformada aproximada de Radon en 4D

Y, equivalentemente, para 4D, definimos,

$$\mathfrak{R}f(s_1, s_2, d_1, d_2) = \int f(u_1, u_2, u_1 s_1 + d_1, u_2 s_2 + d_2) du_1 du_2, \quad (12)$$

y su aproximación discreta,

$$\begin{aligned} & \tilde{f}(s_1 | s_2 | d_1 | d_2) = \\ & \sum_{\mathbf{u}_1 \in \mathbb{Z}_2^n} \sum_{\mathbf{u}_2 \in \mathbb{Z}_2^n} f(\lambda(\mathbf{u}_1) | \lambda(\mathbf{u}_2) | l_{s_1}^n(\mathbf{u}_1) + d_1 | l_{s_2}^n(\mathbf{u}_2) + d_2). \end{aligned} \quad (13)$$

Luego, en la etapa m , la transformada parcial vendrá dada por

$$\begin{aligned} & \tilde{f}^m(\overbrace{s_{1n-m}, \dots, s_{1n-1}}^{\mathbf{s}_1} | \overbrace{v_{1m}, \dots, v_{1n-1}}^{\mathbf{v}_1} | \overbrace{s_{2n-m}, \dots, s_{2n-1}}^{\mathbf{s}_2} | \overbrace{v_{2m}, \dots, v_{2n-1}}^{\mathbf{v}_2} | d_1 | d_2) = \\ & \sum_{\mathbf{u}_1 \in \mathbb{Z}_2^m} \sum_{\mathbf{u}_2 \in \mathbb{Z}_2^m} f(\lambda(\mathbf{u}_1, \mathbf{v}_1) | \lambda(\mathbf{u}_2, \mathbf{v}_2) | l_{\lambda(\mathbf{s}_1)}^m(\mathbf{u}_1) + d_1 | l_{\lambda(\mathbf{s}_2)}^m(\mathbf{u}_2) + d_2), \end{aligned} \quad (14)$$

lo cual conduce al siguiente mapeo.

$$\begin{aligned} & \tilde{f}^{m+1}(\overbrace{s_{1n-m-1}, \dots, s_{1n-1}}^{\mathbf{s}_1} | \overbrace{v_{1m+1}, \dots, v_{1n-1}}^{\mathbf{v}_1} | \overbrace{s_{2n-m-1}, \dots, s_{2n-1}}^{\mathbf{s}_2} | \overbrace{v_{2m+1}, \dots, v_{2n-1}}^{\mathbf{v}_2} | d_1 | d_2) = \\ & \tilde{f}^m(\mathbf{s}_1 | 0, \mathbf{v}_1 | \mathbf{s}_2 | 0, \mathbf{v}_2 | d_1 | d_2) + \\ & \tilde{f}^m(\mathbf{s}_1 | 1, \mathbf{v}_1 | \mathbf{s}_2 | 0, \mathbf{v}_2 | d_1 + s_{1n-m-1} + \lambda(\mathbf{s}_1) | d_2) + \\ & \tilde{f}^m(\mathbf{s}_1 | 0, \mathbf{v}_1 | \mathbf{s}_2 | 1, \mathbf{v}_2 | d_1 | d_2 + s_{2n-m-1} + \lambda(\mathbf{s}_2)) + \\ & \tilde{f}^m(\mathbf{s}_1 | 1, \mathbf{v}_1 | \mathbf{s}_2 | 1, \mathbf{v}_2 | d_1 + s_{1n-m-1} + \lambda(\mathbf{s}_1) | d_2 + s_{2n-m-1} + \lambda(\mathbf{s}_2)) \end{aligned} \quad (15)$$

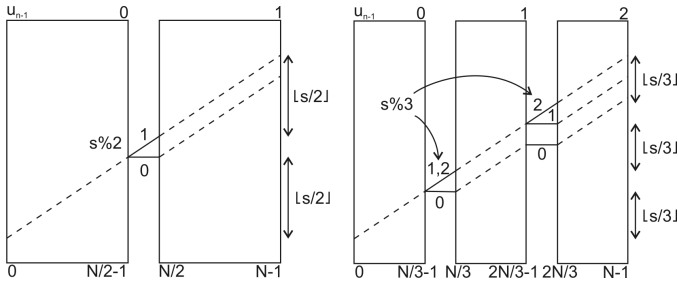


Figura 3. Definición recursiva de líneas discretas por medio de una división a la mitad del dominio y por una división en terceras partes.

III-C. Tamaños no potencias de 2

Las transformadas que hemos definido, ecuaciones (11) y (15) asumen que el tamaño de los datos de entrada es $N \times \dots \times N$, y que N es un número potencia de dos.

Cuando N no es potencia de dos, aún es posible diseñar una transformada aproximada. Pero se requerirá una definición más general de *líneas discretas*, que suplante el papel de la ec. (4), en la cual se basan las transformadas multiescala.

Asumamos que el tamaño de datos será $N = b_0 \times b_1 \times \dots \times b_{n-1}$, y que no necesariamente $b_i = 2$, $\forall i$, tal como era el caso anteriormente: $N = \prod_{i=0}^{n-1} 2 = 2^n$. Si esto es así, un número $u \in [0, N-1]$ se puede descomponer en coeficientes $\mathbf{u} = (u_0, \dots, u_{n-1})$, $u_i \in [0, b_i]$ tales que $\lambda(\mathbf{u}) = \sum_{i=0}^{n-1} \prod_{j=0}^{i-1} b_j u_i$.

Adicionalmente podemos definir un operador de *desplazamiento a derecha*, $u \gg i$, como $\lfloor u / (\prod_{j=0}^{i-1} b_j) \rfloor$, y con ese operador calcular los coeficientes u_i como $u_i = (u \gg i) \% b_i$, donde $\%$ denota el operador resto de la división entera.

Con todo esto en mente, podemos volver sobre la ecuación (4) y observar que es la expresión matemática de la siguiente regla, tal como se ilustra en la figura 3:

si consideramos un dominio dividido en dos mitades de idéntico tamaño, nuestra línea discreta es aquella que cuando alcanza la mitad del dominio, ha llevado a cabo la mitad del ascenso previsto; acerca de cómo actuar dentro de estas dos mitades, aplíquese recursivamente esta regla.

Así, en la ec. (4) el primer sumando, $l_{\lfloor s/2 \rfloor}^{n-1}(u_0, \dots, u_{n-2})$, da cuenta de «dentro de estas dos mitades, aplíquese la misma regla», mientras que el segundo, juega el papel de «llevar a cabo la mitad del ascenso», $\lfloor (s+1)/2 \rfloor$, «cuando se alcanza la mitad del dominio», u_{n-1} . Nótese que $\lfloor (s+1)/2 \rfloor = \lfloor s/2 \rfloor + s_0 = (s \gg 1) + s_0$, y que el término $+s_0$, refleja la posibilidad de que s sea un número impar, en cuyo caso este «resto» debe ser «ascendido» en la frontera entre ambas mitades. Obsérvese también, que esta fórmula relaciona un índice binario en la variable de integración con uno en la pendiente/ascenso. Y que la significancia de los índices relacionados en u y s son opuestas: $u_{n-1-m} \iff s_m$.

Por todo ello, podemos redefinir la ec. (4) de tal forma que siga expresando la misma idea que se plasmó como «regla escrita», pero sin asumir que s_m y u_{n-1-m} han de

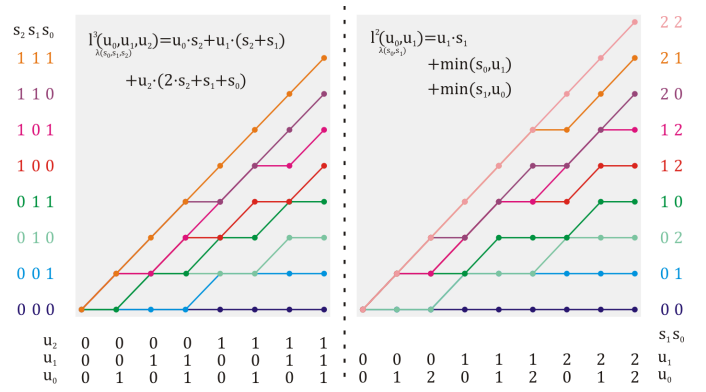


Figura 4. Líneas discretas para $N=8$ y base binaria; y $N=9$ con base ternaria.

ser necesariamente índices binarios.

$$l_s^n(u_0, \dots, u_{n-1}) = l_{s \gg 1}^{n-1}(u_0, \dots, u_{n-2}) + \min(u_{n-1}, s_0) + (s \gg 1)u_{n-1} \quad (16)$$

Como era de esperar, la ec. (4) resulta ser la forma simplificada de (16) para el caso $N = 2^n$. El resto del ascenso, s_0 , que previamente se realizaba en el único cambio entre subdominios, u_{n-1} y se expresaba por la multiplicación de ambos índices binarios, ahora se distribuirá haciendo uso de $\min(u_{n-1}, s_0)$.

IV. CONCLUSIONES

El aparato matemático que hemos presentado es útil para computar el operador de formación fotográfico a partir de una imagen plenóptica (ver detalles en [6]), con solo aplicar lo aquí visto al operador continuo $\mathfrak{R}f(s, d_1, d_2) = \int f(u_1, u_2, u_1 s + d_1, u_2 s + d_2) du_1 du_2$. Esta aplicación se recoge en [7]. Como resultado se obtiene una rebaja típica del 80% del tiempo de cómputo requerido por el algoritmo previo más rápido.

AGRADECIMIENTOS

Este trabajo está parcialmente financiado por el “Programa Nacional de Diseño y Producción Industrial” (Project DPI 2006-07906) del “Ministerio de Ciencia y Tecnología”, y por el “European Regional Development Fund” (ERDF).

REFERENCIAS

- [1] J. Radon, “Über die bestimmung von funktionen durch ihre integralwerte längs gewissermannigfaltigkeiten,” *Ber. Vehr. Sächs. Akad. Wiss. Leipzig. Math. Nat. Kl.*, vol. 69, pp. 262–277, 1917.
- [2] M. L. Brady and W. Yong, “Fast parallel discrete approximation algorithms for the Radon transform,” in *SPAA*, 1992, pp. 91–99.
- [3] M. L. Brady, “A fast discrete approximation algorithm for the Radon transform,” *SIAM J. Comput.*, vol. 27, no. 1, pp. 107–119, 1998.
- [4] W. Götz and H. Druckmüller, “A fast digital Radon transform - an efficient means for evaluating the Hough transform,” *Pattern Recognition*, vol. 29, no. 4, pp. 711–718, April 1996.
- [5] A. Brandt and J. Dym, “Fast calculation of multiple line integrals,” *SIAM J. Sci. Comput.*, vol. 20, no. 4, pp. 1417–1429, 1999.
- [6] R. Ng, “Fourier slice photography,” *ACM Trans. Graph.*, vol. 24, no. 3, pp. 735–744, July 2005.
- [7] J. G. Marichal-Hernández, J. P. Lüke, F. Rosa, F. P. Nava, and J. M. Rodríguez-Ramos, “Fast approximate focal stack transform,” in *3DTV '09*, 2009.