

Capítulo 5

5. El Algoritmo Lbp



Ilustración 1 Imagen obtenida al aplicar el algoritmo LBP a una imagen monoespectral

5.1 Análisis de texturas con patrones locales binarios

La textura en una imagen puede proveer información invaluable al momento de identificar objetos presentes en ésta. El sistema de visión humano no sólo puede distinguir objetos basándose en el color; también puede distinguir la textura.

La principal característica de la textura es la repetición de un patrón o de varios patrones contenidos en una región. El patrón puede repetirse exactamente, o puede comportarse como un conjunto de pequeñas variaciones, esto lo realiza posiblemente como una función de posición. (Lucieer, 2005).

Un simple y nuevo modelo de textura es el LBP (Ojala, 1996), el cual ha demostrado ser un método adecuado para analizar texturas en imágenes monoespectrales. El LBP se ha convertido una medida de la textura de una imagen, mostrando excelentes resultados en términos de precisión y complejidad computacional.

El operador LBP puede ser visto como un acercamiento unificador hacia los modelos estadísticos y los modelos estructurales del análisis de texturas.

En el modelo LBP la textura se describe en términos de micro-primitivas (textones) y sus reglas estadísticas; opcionalmente, las primitivas pueden ser utilizadas como una medida complementaria del contraste local de la imagen, lo cual mide la fuerza de las primitivas.

Al principio el operador LBP de textura fue introducido como una medida complementaria para medir el contraste local de una imagen. En esta primera versión el LBP trabaja con los 8 vecinos más cercanos de un píxel y maneja los conceptos de umbral y pesos. El umbral es un valor que se toma como referencia y es propio de cada vecindad; los pesos son valores resultantes de trabajar con 2^k , donde $0 \leq k \leq 7$.

El valor del píxel central es el que se toma como umbral, posteriormente se produce un código LBP para cada vecindad al multiplicar los valores del umbral con los pesos dados a los píxeles correspondientes y finalmente se suma el resultado.

Para ejemplificar lo anterior se tiene el ejemplo con la siguiente vecindad:

6	5	2
7	6	1
9	8	7

Primero se localiza al píxel central y su valor; para la vecindad del ejemplo este valor es de 6.

Los valores de los píxeles de la vecindad (P_v) se comparan con el valor del píxel central (P_c) y se establece la siguiente condición:

$$\text{Si } \begin{cases} P_v \geq P_c \Rightarrow P_v = 1 \\ P_v < P_c \Rightarrow P_v = 0 \end{cases} \quad (\text{V. 1})$$

Se construye una nueva tabla, con los valores de los píxeles de la vecindad umbralizada

Umbral		
1	0	0
1		0
1	1	1

Se crea una tabla de pesos ($\text{pesos}=2^k, 0 \leq k \leq 7$) en donde la posición de cada uno se asigna arbitrariamente, para el ejemplo se asignó en dirección de las manecillas del reloj, comenzando en la esquina superior derecha.

Pesos		
1	2	4
128		8
64	32	16

Los valores de los pixeles en la vecindad umbralizada son multiplicados por los pesos dados a los pixeles correspondientes

Umbral		Pesos	=	LBP																											
<table border="1"> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td></td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	1	0	0	1		0	1	1	1	X	<table border="1"> <tr><td>1</td><td>2</td><td>4</td></tr> <tr><td>128</td><td></td><td>8</td></tr> <tr><td>64</td><td>32</td><td>16</td></tr> </table>	1	2	4	128		8	64	32	16	=	<table border="1"> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>128</td><td></td><td>0</td></tr> <tr><td>64</td><td>32</td><td>16</td></tr> </table>	1	0	0	128		0	64	32	16
1	0	0																													
1		0																													
1	1	1																													
1	2	4																													
128		8																													
64	32	16																													
1	0	0																													
128		0																													
64	32	16																													

De aquí se obtiene el siguiente patrón binario: 11110001

El patrón se comienza a formar de izquierda a derecha el cual corresponde a los valores del umbral.

Los valores de los 8 pixeles resultantes de multiplicar el umbral por los pesos son sumados para obtener el número LBP del vecindario:

128	64	32	16	8	4	2	1
1	1	1	1	0	0	0	1

$$LBP = 1 + 0 + 0 + 0 + 16 + 32 + 64 + 128 = 241$$

Se obtiene el valor de C (contraste), donde:

$$C = \frac{\text{Promedio de valores sin umbralizar con valor de } 1}{\text{Promedio de valores sin umbralizar con valor de } 0} - \quad (V.2)$$

$$C = \frac{6+7+8+9+7}{5} - \frac{5+2+1}{3} = 4.7$$

El promedio de los valores de los niveles de gris debajo del píxel central es sustraído de todos los valores de los niveles de gris superiores o iguales al valor del píxel central.

Las distribuciones bidimensionales del LBP y las medidas del contraste local fueron usadas como características.

El operador fue nombrado LBP/C y se han reportado muy buenos índices de discriminación con texturas seleccionadas del álbum fotográfico de Brodatz.

El operador LBP está relacionado con métodos muy bien conocidos del análisis de texturas; estas relaciones se muestran en el esquema de la Ilustración 50.

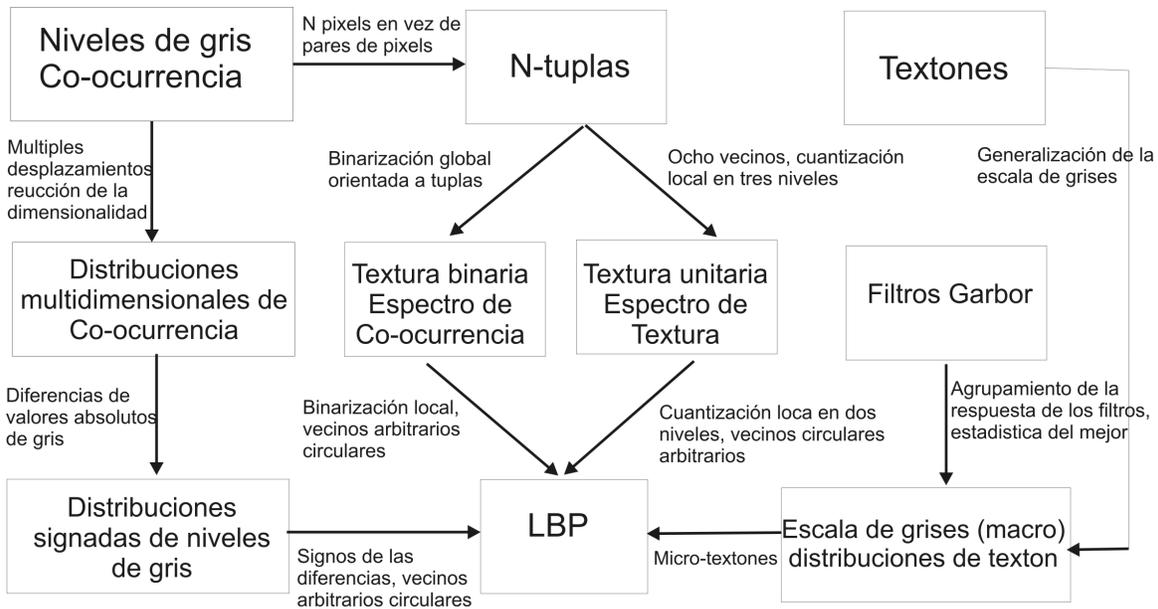


Ilustración 2 Relaciones del operador LBP con otros métodos de textura

5.2 Formalismo teórico

Se define la textura T en un vecindario local de una imágenes en niveles de gris, como la distribución conjunta de los niveles de gris de $P + 1$ ($P > 0$) pixeles en una imagen.

$$T = t(g_c, g_0, \dots, g_{p-1}) \quad (V.3)$$

Donde g_c corresponde al valor de gris del píxel central de un vecindario local y g_p ($p = 0, 1, \dots, p-1$) corresponde a los valores de gris de P igualmente espaciados sobre un círculo de radio R ($R > 0$) que forma un conjunto de vecinos simétrico circular. Este conjunto de $P + 1$ pixeles es después denotado por G_p .

En el dominio digital de una imagen, las coordenadas de los vecinos g_p están dadas por

$$x_c + R \cos \frac{2\pi p}{P}, \quad y_c + R \sin \frac{2\pi p}{P} \quad (V.4)$$

donde (x_c, y_c) son las coordenadas del píxel central

La Ilustración 51 ilustra tres conjuntos de vecinos circulares simétricos con diferentes valores de P y R . Los valores de los vecinos que no caen exactamente en los pixeles son estimados por interpolación lineal. Puesto que la correlación entre pixeles decrece con la distancia, la información textural de una imagen puede ser obtenida de vecindarios locales.

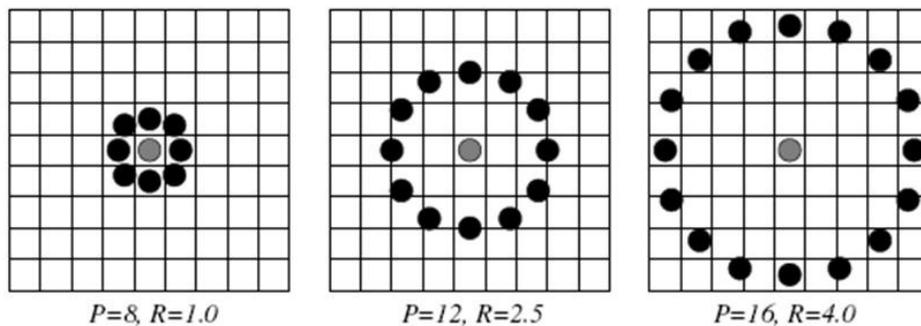


Ilustración 3 Diferentes conjuntos de vecinos circulares simétricos con valores de P y R diferentes

Si el valor del píxel central es restado a los valores de los vecinos, la textura local puede ser representada – sin perder información – como una distribución conjunta del valor del píxel central y las diferencias:

$$T = t(g_c, g_0 - g_c, \dots, g_{p-1} - g_c) \quad (V.5)$$

Suponiendo que las diferencias son independientes de g_c , la distribución puede ser factorizada:

$$T \approx t(g_c)(g_0 - g_c, \dots, g_{p-1} - g_c) \quad (V.6)$$

En la práctica, la suposición de independencia podría no siempre ser verdad. Debido a la naturaleza limitada de los valores en las imágenes digitales muy altos y muy pocos valores de g_c serán obviamente limitados abajo del rango de las posibles diferencias. Sin embargo, aceptando las pequeñas posibles pérdidas de información es posible obtener invariancia con respecto a los cambios en la escala de grises.

Debido a que $t(g_c)$ describe el total de la luminosidad de una imagen, la cual no tiene relación con la textura local de la imagen, $t(g_c)$ no provee información útil para el análisis de textura.

Por tanto, mucha de la información acerca de las características de textura en la distribución conjunta original es preservada en la distribución conjunta de las diferencias

$$T \approx t(g_0 - g_c, \dots, g_{p-1} - g_c) \quad (V.7)$$

La distribución de la diferencia P-dimensional relaciona la ocurrencia de diferentes patrones de textura de cada píxel en el vecindario. Para una constante o lenta variación en las regiones, las diferencias del grupo son cercanas a cero.

En una mancha (*spot*), todas las diferencias son relativamente grandes. Sobre un borde, las diferencias en algunas direcciones son más grandes que en otras.

Aunque invariante contra del cambio en la escala de grises, las diferencias son afectadas al escalarlas. Para conseguir la invariancia respecto a cualquier transformación monótonica en niveles de grises, únicamente se consideran los signos de las diferencias.

$$T \approx t[s(g_0 - g_c), \dots, s(g_{p-1} - g_c)] \quad (\text{V.8})$$

Donde

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (\text{V.9})$$

Ahora bien, pesos binomiales 2^p son asignados a cada signo $s(g_p - g_c)$ transformando las diferencias de un vecindario en un código LBP único. Este código caracteriza la textura local de la imagen alrededor de (x_c, y_c) :

$$\text{LBP}_{p,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p \quad (\text{V.10})$$

En la práctica la ecuación anterior significa que los signos de las diferencias en un vecindario son interpretados como números binarios P-bit, resultando en 2^P distintos valores de código LBP.

La distribución local en niveles de grises, por ejemplo la textura puede ser descrita con una distribución discreta 2^P -bin de código LBP.

$$T \approx t(\text{LBP}_{p,R}(x_c, y_c)) \quad (\text{V.11})$$

Suponiendo que se tiene una imagen dada de dimensión $N \times M$, es decir $(x_c \in \{0, \dots, N-1\}, y_c \in \{0, \dots, M-1\})$. Calculando la distribución $\text{LBP}_{p,R}$ (vector característico) para esta imagen, sólo se considera la parte central porque un vecindario suficientemente grande no puede ser usado en los bordes. El código LBP es calculado para cada píxel en una porción muy corta de la imagen, y la distribución del código es usada como un vector característico, denotado por S :

$$S = t(\text{LBP}_{p,R}(x, y)), x \in \{[R], \dots, N-1-[R]\}, y \in \{[R], \dots, M-1-[R]\} \quad (\text{V.12})$$

El LBP original es muy similar al $LBP_{8,1}$, con dos diferencias. Primero, el vecindario en la definición general es indexado circularmente, haciendo más fácil obtener la rotación invariante de los descriptores de textura. Segundo, los pixeles en la diagonal en un vecindario de 3×3 son interpolados en $LBP_{8,1}$.

Ojalá et. al. (1996) encontró que no todos los patrones locales binarios describen de manera apropiada algunas propiedades de la textura. El algoritmo LBP captura la uniformidad del pixel central en relación con sus vecinos, pero no captura la uniformidad del vecindario por sí mismo.

El concepto de patrones “uniformes” fue introducido por Mäenpää, él observó que ciertos patrones parecen ser fundamentales en propiedades de la textura. Proporcionando una inmensa mayoría de patrones, algunas veces sobre el 90%. Esta proposición fue más allá, ya que fue confirmada con una gran cantidad de datos por Ojala y otros. Estos patrones son llamados “uniformes” porque ellos tienen una cosa en común, al menos existen dos transiciones en el código binario circular: Una transición de uno a cero o una transición de cero a uno.

Para definir formalmente la “uniformidad” de un vecindario G , se necesita una medida de uniformidad U :

$$U(G_p) = |s(g_{p-1}, g_c) - s(g_0, g_c)| + \sum_{p=1}^{P-1} |s(g_p, g_c) - s(g_{p-1}, g_c)| \quad (V.13)$$

Patrones con un valor de U menores o iguales a dos son designados como “uniformes”. Para un número binario P -bit, el valor de U puede ser calculado eficientemente con aritmética binaria como sigue:

$$U(x) = \sum_{p=0}^{P-1} F(x \text{ xor ROR}(x,1), p) \quad (V.14)$$

Donde x es un número binario.

U corresponde al número de transiciones espaciales o cambios de bits de 0/1 en el patrón.

Cuando $g_p = g_0$, U_c (uniformidad del pixel central) es definida como:

$$U_c = \sum_{i=1}^P |\text{sign}(g_i - g_c) - \text{sign}(g_{i-1} - g_c)| \quad (\text{V. 13})$$

5.3 Contraste y patrones de textura

El contraste es una propiedad de la textura usualmente considerada como una sugerencia importante para nuestra visión de sistema, pero el operador LBP por si mismo ignora totalmente la magnitud de las diferencias en los niveles de gris. En muchas aplicaciones, especialmente en la inspección visual industrial, la luminosidad puede ser controlada de manera exacta. En tal situación, una escala de grises puramente invariante al operador de textura podría perder información útil, y adicionalmente la información dependiente de la escala de grises, podría elevar la precisión del método. Además, en aplicaciones tales como la segmentación de una imagen, los cambios graduales en la iluminación podrían no requerir de usar un método invariante en niveles de gris.

En una visión más general, la textura es distinguida no solo por patrones de textura, sino también por la fuerza de los patrones. La textura puede hasta ser incluso considerada como un fenómeno bidimensional caracterizado por dos propiedades ortogonales: estructura espacial (patrones) y contraste (la fuerza de los patrones). La información de los patrones es independiente de la escala de grises, mientras que el contraste no lo es. Por otro lado, el contraste no es afectado por la rotación, pero los patrones si lo son. Estas dos medidas se complementan a cada otra de una manera muy útil.

El operador LBP fue originalmente diseñado para este propósito: complementar una medida de escala de grises dependiente de la “cantidad” de textura. Ojala y otros usan una distribución conjunta del código LBP y una medida del contraste local (LBP/C) como descriptores de textura.

La limitación más importante del operador original LBP es su pequeña área espacial de soporte. Las características calculadas en un vecindario local de 3×3 no pueden capturar estructuras a gran escala que tal vez sean las características dominantes de algunas texturas. Sin embargo, los códigos adyacentes LBP no son totalmente independientes uno del otro.

Como un muy simple operador de textura, el LBP es idealmente apropiado para aplicaciones que requieren una rápida extracción de características. Debido a su simplicidad y ejecución, el operador LBP se ha aplicado en:

- Inspección Visual Industrial
- Recuperación de Imágenes
- Análisis de escenas
- Análisis de rostros, etcétera.

5.4 Modificaciones al algoritmo LBP para imágenes multiespectrales

Para el desarrollo de la presente tesis se tomo del algoritmo LBP el principio de trabajar con los ocho vecinos más cercanos de un pixel y manejar exclusivamente el concepto de umbral, donde nuestro umbral fue el valor del pixel central. También ocupamos el concepto de ventana en la cual están presentes los ocho vecinos del pixel central y la cual se desliza a lo largo y ancho de la imagen digital.

El primero concepto que tomamos en cuenta para adaptar el algoritmo Lbp fue que en un principio este algoritmo fue creado para imágenes monoespectrales, en cambio en la percepción remota estudiamos imágenes multiespectrales, así que ese fue el primer cambio, pasar de una imagen monoespectral a una multiespectral.

Como nuestra imagen multiespectral está formada por siete bandas, cada pixel central es un vector compuesto por siete direcciones P_c ($pc_1, pc_2, pc_3, pc_4, pc_5, pc_6, pc_7$), y los ocho vecinos en consecuencia también son vectores formados por siete direcciones P_{v1} ($pv1_1, pv1_2, pv1_3, pv1_4, pv1_5, pv1_6, pv1_7$), P_{v2} ($pv2_1, pv2_2, pv2_3, pv2_4, pv2_5, pv2_6, pv2_7$), ..., P_{v8} ($pv8_1, pv8_2, pv8_3, pv8_4, pv8_5, pv8_6, pv8_7$). Para obtener un campo vectorial que representara la textura de la imagen se decidió realizar la diferencia del vector pixel central P_{ci} y de cada vector pixel vecino P_{vji} donde $1 \leq i \leq 7$ y representa el numero de bandas de la imagen multiespectral y $1 \leq j \leq 8$ representa el número de vecinos del pixel central

Como resultado de la diferencia del pixel central P_{ci} y el pixel vecino P_{vij} , obtenemos ocho vectores diferencia formados por 7 componentes que corresponden a las bandas de la imagen multiespectral $Vd1_1$ ($d1_1, d1_2, d1_3, d1_4, d1_5, d1_6, d1_7$), $Vd2_1$ ($d2_1, d2_2, d2_3, d2_4, d2_5, d2_6,$

d_{27}), ..., V_{d8_1} ($d_{8_1}, d_{8_2}, d_{8_3}, d_{8_4}, d_{8_5}, d_{8_6}, d_{8_7}$), es decir los vectores diferencia los podemos representar como $V_{dji} = P_{ci} - P_{vji}$.

Como el resultado que obtuvimos está conformado por ocho vectores diferencia, se decidió ocupar el vector que nos dio la máxima diferencia ($V_{\text{maxdif } j}$). Este vector está conformado por siete componentes V_{maxdif} ($V_{\text{maxdif}1}, V_{\text{maxdif}2}, V_{\text{maxdif}3}, V_{\text{maxdif}4}, V_{\text{maxdif}5}, V_{\text{maxdif}6}, V_{\text{maxdif}7}$).

El resultado que deseamos obtener es un realce en los bordes y la textura de la imagen, se decidió entonces, que el vector vecino más representativo es el que nos mostraba la máxima diferencia de los ocho vecinos.

El vector de máxima diferencia presenta un campo vectorial que nos representa la textura en la imagen, pues si en el campo vectorial no hay cambios significativos, es decir no hay diferencias muy grandes la textura de la imagen será lisa o uniforme, en cambio, si el campo vectorial presenta grandes diferencias la textura de la imagen será rugosa.

El vector máxima diferencia nos representa la textura de la vecindad, por lo tanto el pixel central pasa a ser sustituido por el vector máxima diferencia. Cuando se termina este proceso de analizar los vectores la ventana deslizante pasa al pixel vecino contiguo y repite el mismo proceso hasta llegar a recorrer la imagen multiespectral completa.

La metodología que seguimos fue primero realizar un programa en MATLAB para el análisis de imágenes multiespectrales. Su diseño y construcción está asociado con el número de bandas de la imagen multiespectral. Para imágenes Landsat TM se considera una dimensión de 6 bandas ya que no se incluye la banda del infrarrojo térmico. También se puede reducir el número de dimensiones a un tamaño menor al aplicar la descomposición en componentes principales a la imagen multiespectral, en tal caso el programa solo analizaría 3 bandas en donde la correlación entre éstas es de cero. Para otras imágenes multiespectrales la dimensionalidad puede ser diferente, por todo lo anterior, el algoritmo que se desarrolla en esta tesis emplea un número de N bandas multiespectrales.

Hay que mencionar que el resultado del algoritmo arroja también una imagen multiespectral, es decir el número de bandas de entrada es igual al número de bandas de salida.

Si se carga una imagen monoespectral, el algoritmo procesa con éxito su única banda, dando como resultado una imagen monoespectral.

El procedimiento de este algoritmo, resultado de modificar el algoritmo LBP para imágenes multiespectrales, se describe a continuación:

- a) Se carga la imagen multiespectral de N bandas (de 1 a 10) en memoria principal. Estas bandas pueden ser las bandas originales o bien bandas que hayan sido modificadas por algún tipo de transformación, esto lo decide el usuario. En el capítulo VI se proponen diferentes combinaciones para observar cuál es la combinación que proporciona el mejor resultado.
- b) Se determina la dimensión de la imagen multiespectral, es decir alto (Y) ancho (X), y número de bandas (N)
- c) Se asignan las coordenadas de los 8 vecinos más cercanos

$$[-1 -1; -1 0; -1 1; 0 -1; 0 1; 1 -1; 1 0; 1 1]$$
- d) Se localiza el origen la ventana inicial que tiene un tamaño de $3 \times 3 \times N$
- e) Se recorre la imagen desde la esquina superior derecha en el pixel que corresponde a la coordenada (2,2,N) de la imagen. Al momento que la imagen es recorrida por la ventana se resta el valor del centro de la ventana menos el valor de los vecinos
- f) Se realiza una nueva imagen resultado, la cual tiene una dimensión de (X-2,Y-2,N) la cual contiene el valor máximo resultado de la resta de los centros menos los vecinos, esta imagen contiene un mapa de textura de la imagen.
- g) La imagen resultado es al igual que la imagen original una imagen multiespectral, que está compuesta por el vector de máximas diferencias.