

UNIVERSITÀ DEGLI STUDI DI FIRENZE

Laurea Magistrale in Ingegneria Informatica

Corso di Analisi di Immagini e Video

Implementazione Multiscale Local Binary Pattern

A. Rizzo, M. Bruni

Anno accademico 2012/2013

Indice

1	Introduzione	2
1.1	Description of region features	2
2	Multi-scale Local Binary Pattern	3
2.1	Local Binary Pattern	3
2.1.1	Descrizione	3
2.2	Extended LBP	5
2.3	Uniform Local Binary Pattern	6
2.4	Multiscale Local Binary Pattern	8
2.4.1	Descrittore	9
2.5	Filtro di smoothing gaussiano	9
3	Implementazione	11
3.1	Segmentazione	11
3.2	Misura di similarità	11
4	Conclusioni	15
	Acronimi	16
	Bibliografia	17

1 Introduzione

1.1 Description of region features

Texture is a very general notion that can be attributed to almost everything in nature. For a human, the texture relates mostly to: A specific, spatially repetitive (micro)structure of surfaces formed by repeating a particular element or several elements in different relative spatial positions. Generally, the repetition involves local variations of scale, orientation, or other geometric and optical features of the elements. Texture is a property of regions: the texture of a point is undefined. Texture involves the spatial distribution of gray levels or colors. Texture in an image can be perceived at different scales or levels of resolution. A region is perceived to have texture when the number of primitive elements in the region is large. If only a few primitive elements are present, then a group of countable objects is perceived instead of a textured image. Texture. It is almost impossible to provide a complete description of textures in words. However, words capture some informal qualitative features that can help discriminate different textures: fineness - coarseness, smoothness, granularity, lineation, directionality, regularity - randomness.

Texture models can be of interest for three main purposes: Segmentation: partition the input image into regions of uniform texture.

Classification: produce a classification map of the input image where each uniform textured region is identified with the texture class it belongs to.

Synthesis: generate an image with a texture that looks like a requested texture class.

2 Multi-scale Local Binary Pattern

In questo capitolo verranno presentati alcuni concetti teorici e gli strumenti utilizzati per lo sviluppo dell'applicazione. Il primo paragrafo introdurrà il descrittore di feature Local Binary Pattern (LBP). Verrà analizzata una sua variante, ovvero Uniform LBP, che viene utilizzato per ridurre la lunghezza del vettore di feature. Analizzeremo infine la sua estensione Multi-scale Local Binary Pattern (MLBP).

2.1 Local Binary Pattern

Local Binary Pattern (LBP) è un efficiente texture operator, introdotto per la prima volta nel 1994 [1]. Le feature estratte con questo metodo vengono utilizzate per la classificazione di immagini in ambito della computer vision. LBP associa ai pixel dell'immagine un'etichetta, determinata tramite un'operazione di confronto tra il pixel in esame ed un certo numero di pixel nelle sue vicinanze, *neighborhood*. Tra le caratteristiche principali di LBP troviamo:

- è robusto alla variazione uniforme dell'illuminazione dell'immagine;
- è robusto al rumore;
- non è invariante alle rotazioni dell'immagine;
- non è invariante alla scala.

Un'altra importante proprietà è infine la sua semplicità computazionale che lo rende ideale per l'analisi delle immagini in Real-Time (RT).

2.1.1 Descrizione

L'idea alla base dell'operatore Local Binary Pattern (LBP) è che le texture bidimensionali possono essere descritte attraverso pattern locali. L'implementazione base dell'operatore LBP etichetta i pixel dell'immagine attraverso un'operazione di sogliatura dei pixel contenuti in una maschera 3×3 .

Il pixel centrale viene confrontato con ogni vicino ed il risultato del confronto viene espresso con un numero binario.

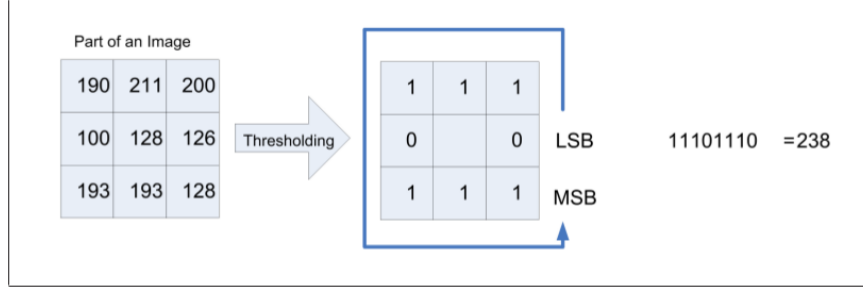


Figura 1: Estrazione del codice LBP.

Riferendoci alla figura 1 il processo di etichettatura dei pixel è il seguente. Il livello di grigio del pixel centrale, detto anche *pivot* viene confrontato con il livello di grigio di tutti gli altri pixel della maschera che viene binarizzata attraverso la seguente funzione:

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

dove x è la differenza tra l'intensità del livello di grigio di un pixel del *neighborhood* ed il livello di grigio del pixel pivot. I valori così ottenuti vengono concatenati in senso antiorario andando a formare un codice binario di 8 bit la cui conversione decimale rappresenta il codice LBP del pixel pivot.

Descrittore

Il numero totale di codici LBP è pari a $2^8 = 256$. L'istogramma dei codici LBP calcolati sui pixel dell'immagine può essere utilizzato come descrittore della texture. In figura 2 è mostrato l'istogramma di un immagine LBP.

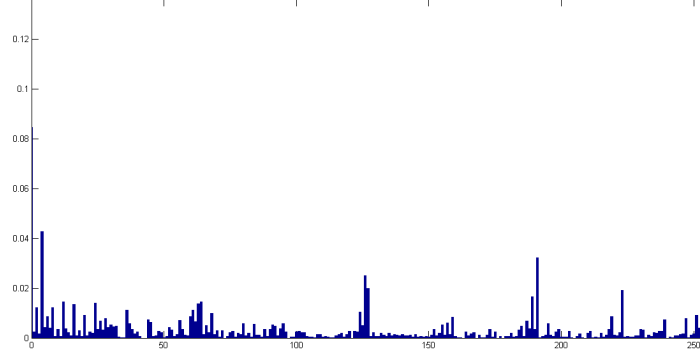


Figura 2: Istogramma normalizzato di un'immagine LBP.

Per il calcolo dell'istogramma si utilizza la seguente formula:

$$h(i) = \sum_{x,y} B(LBP_{P,R}(x,y)) = i, i \in [0, 2^P - 1] \quad (2)$$

dove

$$B(v) = \begin{cases} 1, & v = \text{true} \\ 0, & \text{altrimenti} \end{cases} \quad (3)$$

2.2 Extended LBP

Finora abbiamo considerato come neighborhood i pixel adiacenti a quello di pivot. Una variante dell'operatore LBP è l' Extended LBP (ELBP) o Circular LBP (CLBP) . ELBP considera come neighborhood i P pixel che si trovano ad una distanza R dal pivot. Le coordinate dei P pixel si possono ottenere con la seguente formula:

$$(x_p, y_p) = (-R \sin(\frac{2\pi p}{P}), R \cos(\frac{2\pi p}{P})), \quad p = 0, \dots, P-1 \quad (4)$$

Nel caso in cui le coordinate ottenute non corrispondono alla griglia discreta dell'immagine si effettua una interpolazione bilineare.

In figura 3 sono mostrati alcuni risultati dell'applicazione della formula appena descritta.

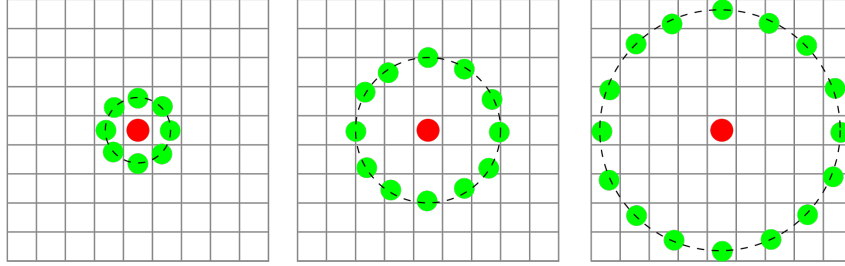


Figura 3: Neighborhood per il calcolo del codice LBP nei casi: $(P = 8, R = 1)$, $(P = 12, R = 2)$ e $(P = 16, R = 4)$.

La formula seguente permette di calcolare, in base dieci, il codice LBP di lunghezza P e raggio R generico di un pixel.

$$LBP_{P,R}(x, y) = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p \quad (5)$$

2.3 Uniform Local Binary Pattern

Come accennato precedentemente l'istogramma dell'immagine LBP è utilizzato come descrittore della texture. Utilizzando l'operatore LBP base, otteniamo un istogramma con 256 classi. Per ridurre la dimensione del descrittore si può utilizzare Uniform Local Binary Pattern (LBP). Uniform LBP considera solo i pattern in cui occorrono al più due transizioni da 0 a 1 o da 1 a 0 tra pixel adiacenti. Per esempio:

- 00000000 \rightarrow 0 transizioni
- 01110000 \rightarrow 2 transizioni
- 11001111 \rightarrow 2 transizioni
- 11001001 \rightarrow 4 transizioni
- 01010010 \rightarrow 6 transizioni

I pattern che soddisfano le condizioni di Uniform LBP sono in totale 58, come mostrato in figura 4.

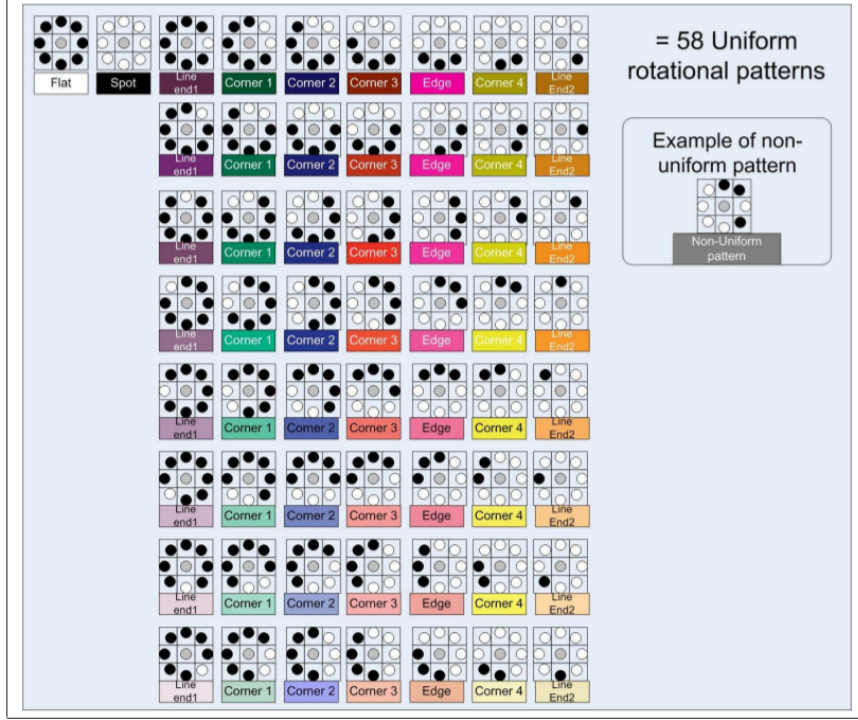


Figura 4: Pattern di Uniform LBP.

Questa variante si basa sul fatto che le feature così estratte sono più robuste al rumore e l'istogramma risultante è di dimensione inferiore mantenendo comunque le feature rilevanti.

In figura 5 è mostrato l'istogramma di un immagine su cui è stato applicato l'operatore di Uniform LBP.

Per ottenere il codice di Uniform LBP si utilizza la formula seguente:

$$LBP_{P,R}^{u2}(x,y) = \begin{cases} I(LBP_{P,R}(x,y)), & \text{se } U(LBP_{P,R}) \leq 2, I(z) \in [0, (P-1)P+2) \\ (P-1)P+2, & \text{altrimenti} \end{cases} \quad (6)$$

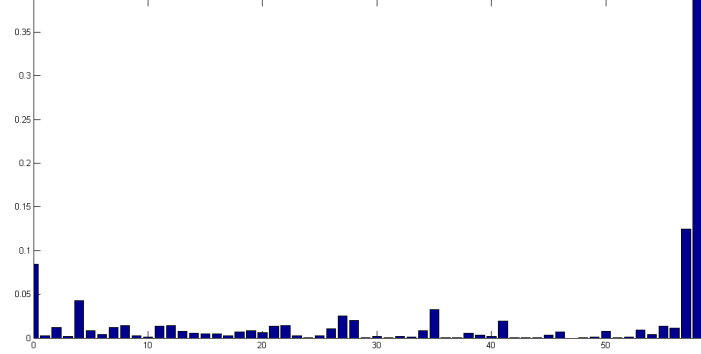


Figura 5: Istogramma normalizzato di una immagine su cui è stato applicato l'operatore di Uniform LBP. I primi 58 pattern sono quelli uniformi, mentre l'ultimo pattern corrisponde a tutti i pixel con codice non uniforme.

dove la funzione $U(x)$ calcola il numero di transizioni.

$$U(LBP_{P,R}) = |s(g_{P-1} - g_c) - s(g_0 - g_c)| + \sum_{p=1}^P |s(g_p - g_c) - s(g_{p-1} - g_c)| \quad (7)$$

2.4 Multiscale Local Binary Pattern

Multi-scale Local Binary Pattern (MLBP) è un'estensione dell'operatore LBP. MLBP è ottenuto combinando i descrittori delle texture ottenuti facendo variare il raggio $r \in [r_{min}, R]$ per la determinazione dei neighbor. Alternativamente MLBP può essere ottenuto applicando l'operatore LBP con raggio costante su l'immagine a risoluzione ridotta. Quest'ultimo metodo è meno efficace in quanto riducendo la risoluzione dell'immagine risulta più difficile estrarre informazioni sul contrasto tra piccole regioni lontane fra loro.

Le feature estratte con MLBP risultano migliori rispetto a LBP per la classificazione di immagini.

2.4.1 Descrittore

Il descrittore della texture ottenuto con MLBP è la concatenazione degli istogrammi ottenuti applicando iterativamente LBP sulla stessa immagine al variare del raggio $r \in \{r_1, r_2, \dots, r_R\}$. Il descrittore è dato dalla seguente formula:

$$f = [h_{P,r_1}, h_{P,r_2}, \dots, h_{P,r_R}] \quad (8)$$

dove

$$h_{P,r}(i) = \sum_{x,y} B(LBP_{P,r}(x,y)) = i, i \in [0, L-1], r \in \{r_1, r_2, \dots, r_R\} \quad (9)$$

con L numero massimo di classi dell'istogramma e

$$B(v) = \begin{cases} 1, & v = \text{true} \\ 0, & \text{altrimenti} \end{cases} \quad (10)$$

2.5 Filtro di smoothing gaussiano

I filtri di *smoothing* vengono utilizzati principalmente per ridurre il rumore presente nell'immagine. Sono anche detti *filtri di media*. Infatti facendo scorrere lungo l'immagine la maschera del filtro, detta anche *kernel mask*, il valore di ogni pixel viene sostituito con la media pesata dei livelli di grigio dei pixel interni alla regione della maschera. Il processo di smoothing permette di ridurre i dettagli meno significativi dell'immagine e mettere in risalto le caratteristiche strutturali della stessa. Un effetto indesiderato dello smoothing è quello di produrre un'immagine sfocata, soprattutto se applicato iterativamente. La figura 6 mostra un filtro di smoothing generico 3x3. La maschera del filtro verrà moltiplicata per un coefficiente di normalizzazione in modo tale che la somma degli elementi della maschera sia uno. I coefficienti della maschera vengono scelti secondo il seguente principio: il peso associato al pixel centrale assume

un valore superiore rispetto agli altri. I pesi associati agli altri pixel assumono valori decrescenti all'aumentare della distanza dal pixel centrale. In altre parole si vuole fare in modo che il pixel centrale abbia una maggiore importanza nel calcolo della media. Viceversa i pixel più lontani da quello centrale peseranno di meno nel calcolo della media. Questo permette di ridurre l'effetto indesiderato del processo di smoothing.

$\omega_{1,1}$	$\omega_{1,2}$	$\omega_{1,3}$
$\omega_{2,1}$	$\omega_{2,2}$	$\omega_{2,3}$
$\omega_{3,1}$	$\omega_{3,2}$	$\omega_{3,3}$

Figura 6: Esempio di una kernel mask di dimensioni 3x3, dove $\omega_{i,j}$ rappresenta il peso applicato al valore del livello di grigio del corrispettivo pixel nell'immagine

I coefficienti della maschera sono calcolati utilizzando la *2D Gaussian Smoothing Operator* $G(x, y)$, cui formula è riportata qui di seguito:

$$G(x, y) = e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (11)$$

La dimensione n della kernel mask è legata al valore di σ dalla seguente formula:

$$6\sigma - 1 = n \quad (12)$$

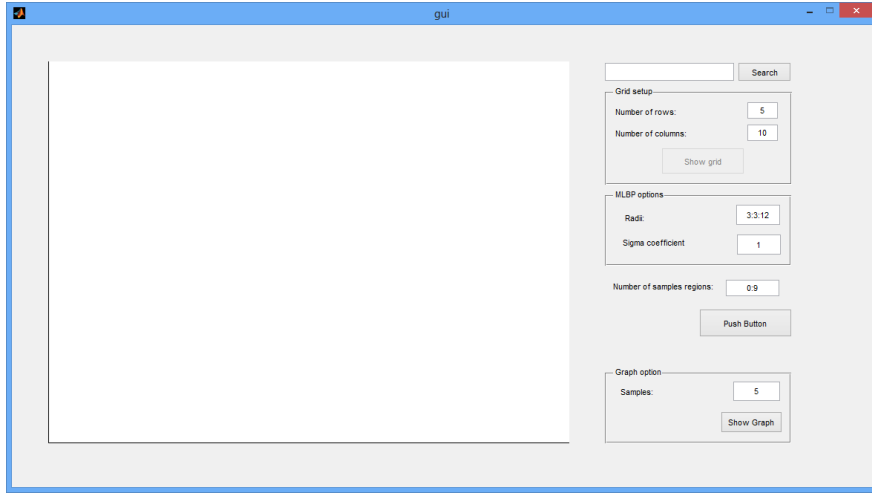


Figura 7: GUI

3 Implementazione

3.1 Segmentazione

3.2 Misura di similarità

Per misurare la similarità tra i descrittori di due regioni I e J , possono essere utilizzati vari criteri. Abbiamo implementato le seguenti misure di similarità $Sim(I, J)$:

- Chi-square criterion:

$$Sim(I, J) = - \sum_i \frac{(f_I(i) - f_J(i))^2}{f_I(i) + f_J(i)} \quad (13)$$

- Histogram intersection:

$$Sim(I, J) = \sum_i \min(f_I(i), f_J(i)) \quad (14)$$

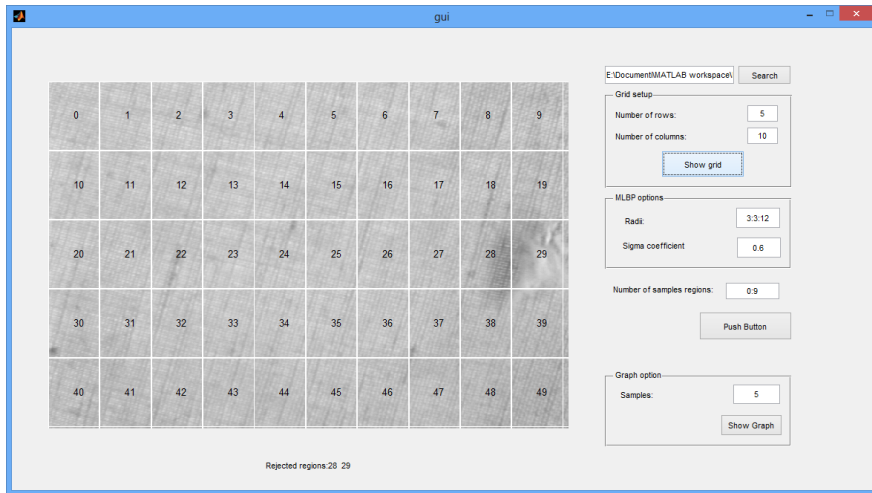


Figura 8: GUI pre LBP

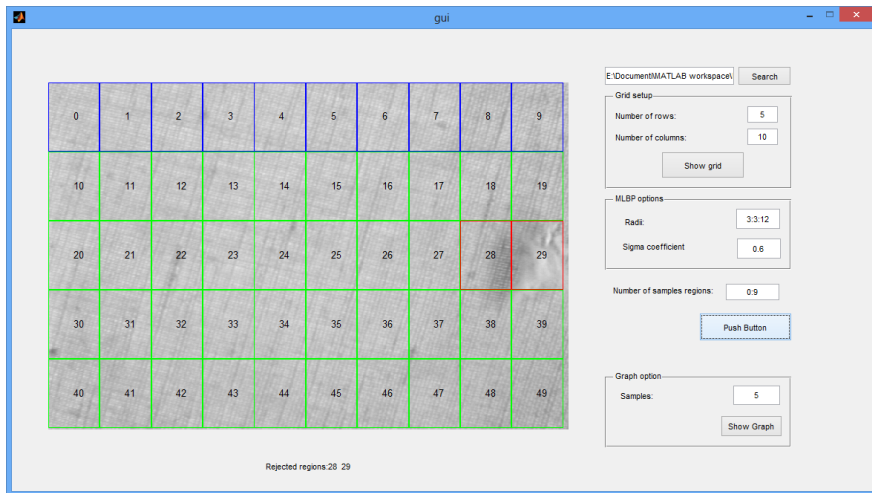


Figura 9: GUI post LBP

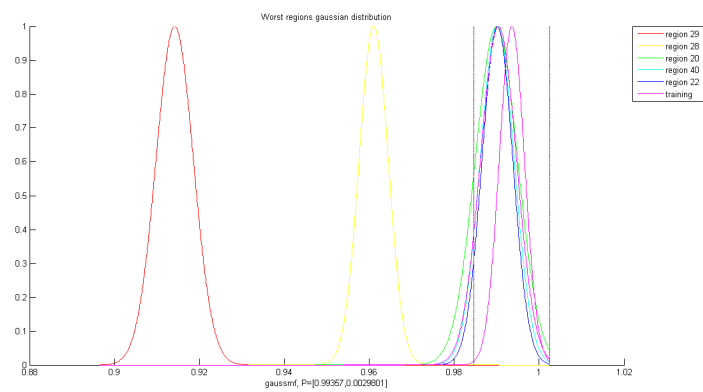


Figura 10: Caso peggiore

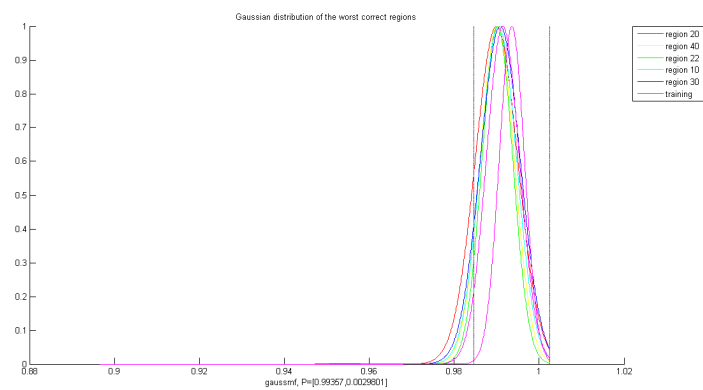


Figura 11: I peggiori tra i migliori

- Log-likelihood ratio (Kullack-Leibler divergence):

$$Sim(I, J) = - \sum_i f_I(i) \log(f_J(i)) \quad (15)$$

- Normalize Correlation:

$$Sim(I, J) = \frac{f_I f'_J}{||f_I|| ||f_J||} \quad (16)$$

- Normalize Cross-correlation:

$$Sim(I, J) = \frac{\sigma_{f_I f_J}}{\sigma_{f_I} \sigma_{f_J}} \quad (17)$$

j = regione r = raggio P = num neighbor

$$h_{P,r,j}(i) = \sum_{x',y' \in M_j} B(LBP_{P,r}(x', y')) = i, i \in [0, L-1], r \in [1, R], j \in [0, J-1] \quad (18)$$

$$B(v) = \begin{cases} 1, & v = \text{true} \\ 0, & \text{altrimenti} \end{cases} \quad (19)$$

$$f_j = [h_{P,1,j}, h_{P,2,j}, \dots, h_{P,R,j}] \quad (20)$$

4 Conclusioni

Gli autori

M. Bruni
A. Rizzo

Acronimi

MLBP Multi-scale Local Binary Pattern

LBP Local Binary Pattern

RT Real-Time

ELBP Extended LBP

CLBP Circular LBP

Elenco delle figure

1	Estrazione del codice LBP.	4
2	Istogramma normalizzato di un'immagine LBP.	5
3	Neighborhood per il calcolo del codice LBP nei casi: ($P = 8, R = 1$), ($P = 12, R = 2$) e ($P = 16, R = 4$).	6
4	Pattern di Uniform LBP.	7
5	Istogramma normalizzato di una immagine su cui è stato applicato l'operatore di Uniform LBP. I primi 58 pattern sono quelli uniformi, mentre l'ultimo pattern corrisponde a tutti i pixel con codice non uniforme.	8
6	Esempio di una kernel mask di dimensioni 3x3, dove $\omega_{i,j}$ rappresenta il peso applicato al valore del livello di grigio del corrispettivo pixel nell'immagine	10
7	GUI	11
8	GUI pre LBP	12
9	GUI post LBP	12
10	Caso peggiore	13
11	I peggiori tra i migliori	13

Elenco delle tabelle

Riferimenti bibliografici

- [1] T. Ojala, M. Pietikainen, and D. Harwood, "Performance evaluation of texture measures with classification based on kullback discrimination of distributions," in *Pattern Recognition, 1994. Vol. 1 - Conference A: Computer Vision and Image Processing., Proceedings of the 12th IAPR International Conference on*, vol. 1, 1994, pp. 582–585 vol.1.