

# COMPUTER VISION

---

UNIVERSITA' DEGLI STUDI DI CATANIA

FACOLTA' DI SCIENZE MATEMATICHE FISICHE NATURALI

CORSO DI LAUREA IN SPECIALISTICA INFORMATICA

A.A. 2006/2007

---

Autore: Carmelo Pulvirenti.

[pulvirenti.carmelo@libero.it](mailto:pulvirenti.carmelo@libero.it)

---

## INTRODUZIONE AL PROGETTO

---

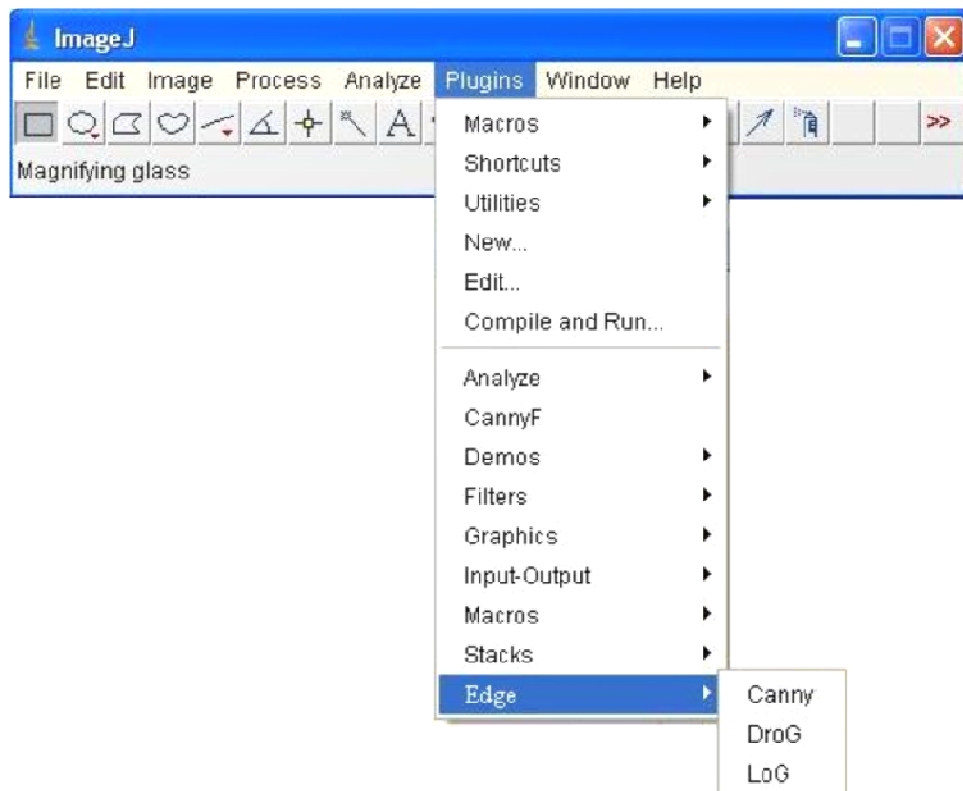
Il progetto realizzato rappresenta un plug-in per ImageJ, un software open source per l'elaborazione delle immagini digitali.

L'applicazione implementa un insieme di tecniche per la detection di edge all'interno di immagini digitali, mediante gli algoritmi di:

- Canny
- DroG
- Laplacian of Gaussian.

Per utilizzare il software in esame è necessario avere installato imageJ ed il plug-in in visione (a tal fine è opportuno rifarsi alla documentazione) nel computer di esecuzione.

Fatte queste operazioni preliminari l'utilizzatore si troverà ad usufruire di un software così composto:

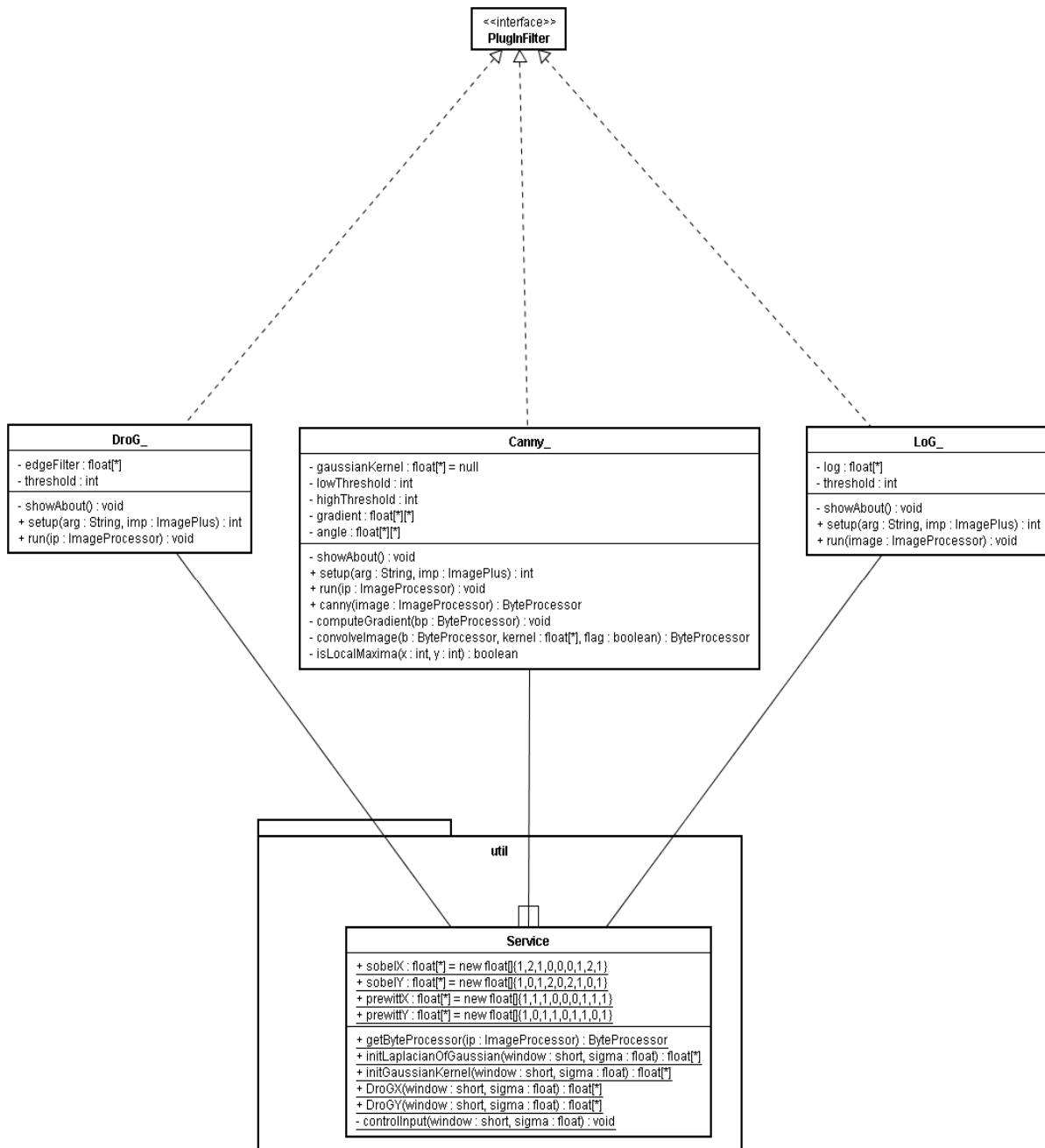


## LE FASI DI PROGETTAZIONE

Le fasi di progettazione hanno dato luogo ad un struttura organizzata su due livelli:

- Edge: un package con l'insieme delle classi che si interfacciano con l'utente.
- Util: un package che offre una serie di servizi per l'implementazione delle tre tecniche atte all'estrazione dei lati.

Tale struttura ha dato luogo al seguente diagramma delle classi sotto indicato:



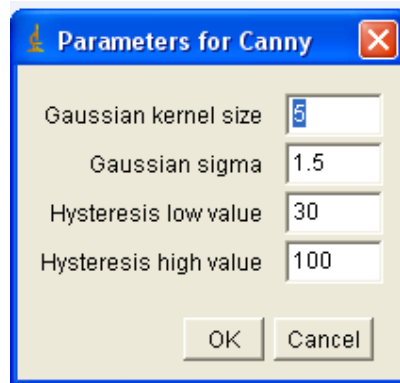
Di seguito saranno analizzate le classi di cui sopra, per fornire al lettore una più semplice visione del lavoro prodotto.

---

## CANNY

---

Rappresenta un operatore composto da più fasi, come indicato dall'autore stesso. La schermata che si presenterà all'utente, dopo aver avviato dal menù plug-in la voce Canny, sarà la seguente:



Attraverso la schermata di cui sopra sarà possibile indicare:

- La grandezza del kernel gaussiano con il relativo valore di sigma.
- Un range di valori compresi tra 0-255 mediante i parametri di low ed high threshold.

I passi computazione sono:

1. Smussare l'immagine tramite l'applicazione della gaussiana costruita mediante i dati forniti dall'utente.
2. Calcolo della magnitudo e della direzione del gradiente. Dall'immagine del punto 1 si applicano due operatori: SobelX e SobelY, ottenendo due immagini dalle quali sarà possibile ricavarne i valori ricercati mediante le formulazioni consuete.
3. Per ogni punto che è un massimo locale, per magnitudo, secondo la direzione dal gradiente, si testerà se:
  - a. Supera high threshold: per cui rappresenta un punto di un lato nell'immagine di output.
  - b. Supera il valore di low threshold ma non quello di high threshold: sarà memorizzato come possibile punto appetente ad un lato.
4. Per tutti i punti trovati al passo 3.b si verificherà se questi sono contigui a edge precedentemente individuati, in tale ipotesi il punto trovato farà parte di un lato, altresì il punto individuato sarà considerato un falso edge per cui sarà scartato dall'immagine di output.

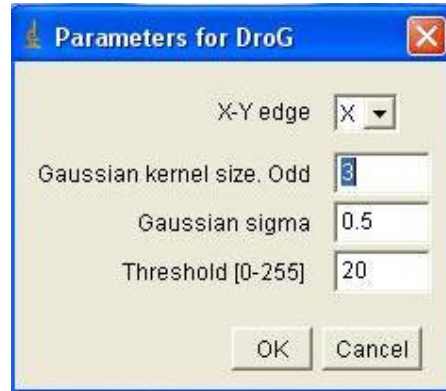
---

## DROG

---

Rappresenta l'operatore per l'estrazione dei lati conosciuto come Derivata della Gaussiana.

La schermata che si presenterà all'utente, dopo aver avviato dal menù plug-in la voce DroG, sarà la seguente:



All'interno di questa sarà possibile indicare:

- La direzione ove applicare l'operatore.
- La grandezza del kernel gaussiano, con il relativo valore di sigma.
- Un valore di Threshold, utilizzato per binarizzare il risultato ottenuto dall'operatore.

I passi dall'algoritmo sono:

1. Costruzione del kernel, secondo la formulazione indicata dall'utente:

$$\text{DroGY}(x,y) = -\frac{x}{\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

$$\text{DroGX}(x,y) = -\frac{y}{\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

A tal scopo si ricorre alle classe Util di Service.

2. Convoluzione dell'immagine con il kernel indicato, e visualizzazione dell'output.
3. Costruzione della LUT mediante il valore di threshold indicato dall'utente.
4. Applicazione all'immagine ottenuta al punto 2, la LUT costruita al passo 3 e visualizzazione dell'output.

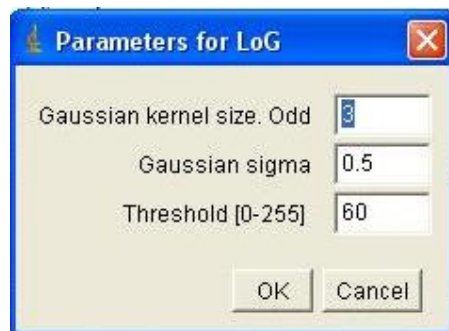
---

## LOG

---

E' l'operatore per l'estrazione dei lati conosciuto come Laplaciano della Gaussiana.

La schermata che si presenterà all'utente, dopo l'avvio dal menù plug-in della voce LoG, sarà la seguente:



Tramite questa iterazione con l'utente sarà possibile indicare:

- La grandezza del kernel gaussiano con il relativo valore di sigma.
- Un valore di threshold utile per il processo di binarizzazione e successivamente per la fase di zero crossing.

I passi computazione di maggior rilievo sono:

1. Calcolo mediante, la classe Util del package service, del kernel di LoG secondo i parametri indicati dall'utente.
2. Convoluzione tra il kernel del punto 1 e l'immagine in input. Il risultato proposto sarà visualizzato come una nuova immagine.
3. Costruzione della LUT (tramite il valore threshold )e convoluzione con l'immagine del punto 2, l'output sarà visualizzato in una nuova immagine.
4. Ricerca degli zero crossing e visualizzazione dell'output ottenuto in una nuova finestra.