



Oil Spill's Features Extraction

Progetto di Image Processing

Vito Domenico Tagliente

Pietro Tota

Luca Riccardi

1. Introduzione

Nel presente elaborato vengono proposte le metodologie e le tecniche che sono state adottate per la realizzazione di un sistema automatico in grado svolgere il compito di “Features extraction” (estrazione delle caratteristiche) nel campo dell’oil spill detection.

Questa fase del processo elaborativo delle immagini è molto importante in quanto permette con facilità, fornito un insieme di immagini da cui apprendere, di estrarre tutte le informazioni possibili da queste macchie di petrolio e di adoperarle per la definizione di un modello. Tale modello verrà in seguito adoperato per la classificazione di successivi dataset di immagini al fine di, definita la conoscenza di base del modello, distinguere le immagini contenenti macchie di petrolio dai falsi positivi, ovvero immagini contenenti macchie “lookalike”, simili.

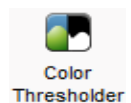
2. Segmentazione

La fase di Segmentazione è stata tralasciata dal contesto di studio per la realizzazione di questo progetto, pertanto è stato possibile semplificare l’operazione concentrando la forza lavoro del team per lo studio e la ricerca delle features.

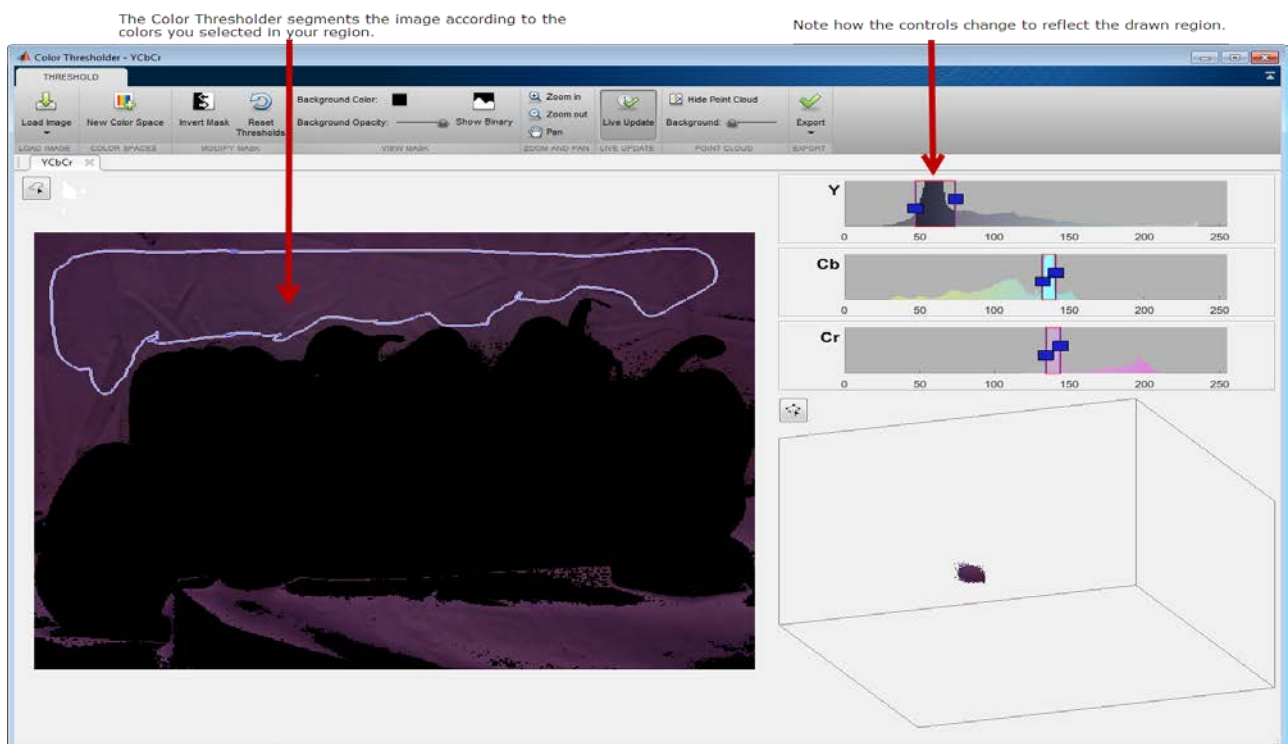
Come già noto, le tecniche di segmentazione possono essere di due tipologie:

- Manuali, basate sull’esperienza dell’operatore
- Automatizzate

In questo contesto è stato pensato di definire una segmentazione manuale basata sull’utilizzo degli strumenti di elaborazione di immagini che Matlab fornisce, il Color Thresholder.



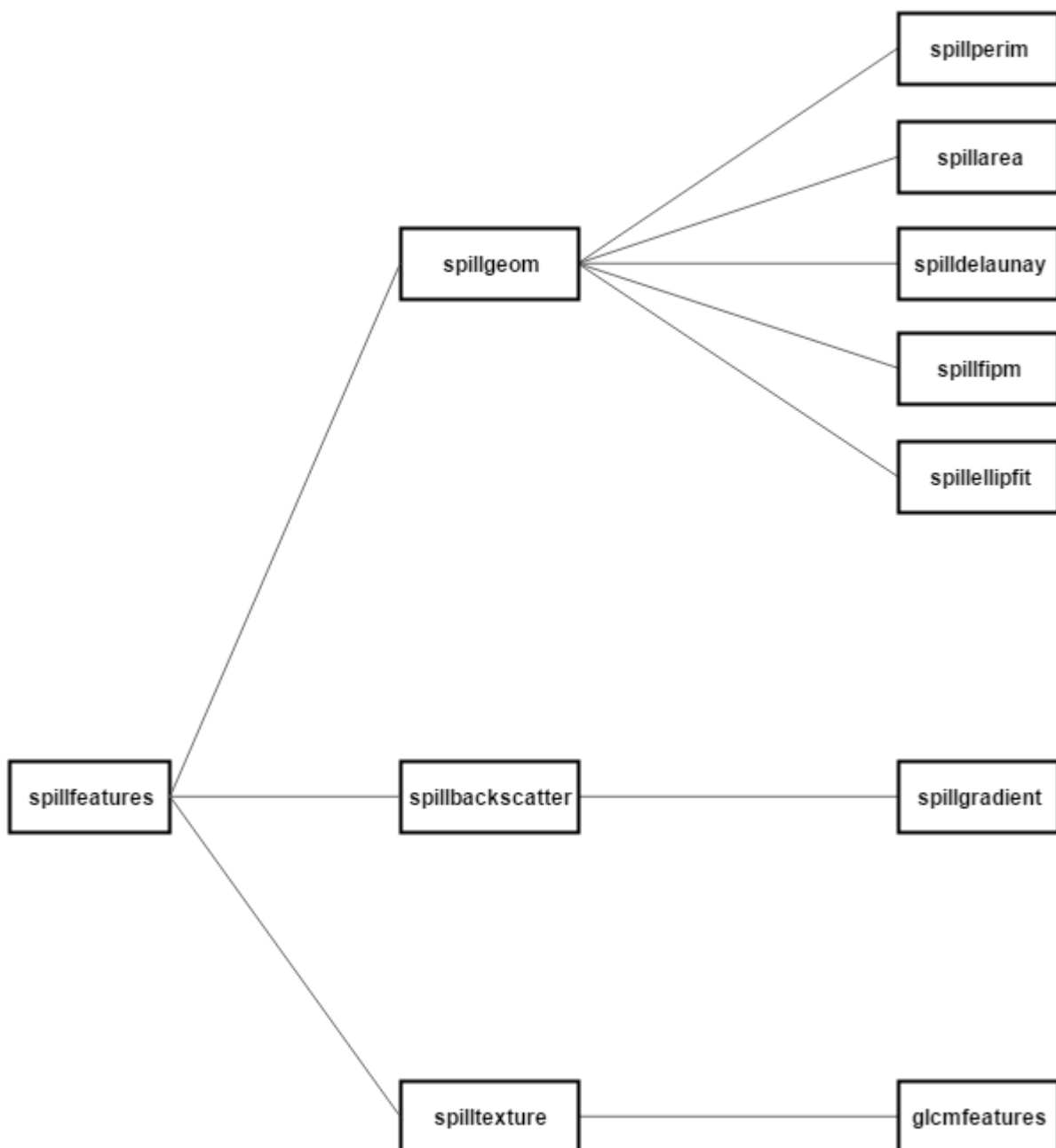
Tramite l’utilizzo di questo toolbox, è stato possibile, adoperando delle operazioni manuali di selezione delle aree dello sfondo di segmentare l’immagine, nel nostro caso la macchia di petrolio.



L'efficienza del tool proposto consiste nella possibilità di convertire la sequenza di operazioni svolte in modalità manuale in uno script, riutilizzabile per studi successivi sulla stessa immagine.

3. Features extraction

Per la realizzazione del progetto in esame, si è pensato di suddividere la complessità del problema in modo tale da isolare il codice per categoria (features Geometriche, Backscatter, Texture) e soprattutto per rendere la struttura del progetto modulare e garantirne la manutenibilità.



Il grafico sopra mostrato rappresenta le relazioni che intercorrono tra i vari script definiti in Matlab, in particolare si evidenziano visivamente le tre macro categorie di features sopra citate:

- Geometriche
- Backscatter
- Texture

4. Preparazione del Workspace

Esaminiamo, ora, le istruzioni preliminari che l'algoritmo realizzato richiede prima di svolgere le proprie elaborazioni. Tale fase, quindi, si occupa di caricare nel workspace di Matlab le variabili che verranno successivamente processate.

Tali istruzioni sono facilmente richiamabili utilizzando lo script ">> *spillbegin*", nel dettaglio otteniamo il seguente listato:

```
% Utilizza questo script per preparare il Workspace di Matlab

% img: immagine originale
img = imread( 'oil.bmp' );
% gimg: immagine non segmentata in scala di grigi
gimg = rgb2gray( img );
% s: immagine segmentata
s = spillseg( img );

% back: immagine di background
back = spillback( img, s );
% gback: immagine di background in scala di grigi
gback = rgb2gray( back );
% gspill: macchia di petrolio in scala di grigi
gspill = gimg - gback;
```

5. Estrazione delle caratteristiche

La fase di estrazione delle features, per come è stato modellato e strutturato il progetto, prevede l'utilizzo di un solo comando "f = spillfeatures(gspill, gback, s)". Tale funzione, dopo aver elaborato i dati di input, ritorna una struttura dati in cui sono presenti tutte le features calcolate.

```
>> spillbegin
>> f = spillfeatures( gspill, gback, s )

f =

    Geometrical: [1x1 struct]
    Backscatter: [1x1 struct]
    Texture: [1x1 struct]

>> f.Geometrical

ans =

    Perimeter: 11070
    Area: 1.6175887500000000e+06
    Complexity: 2.455322988115242e+00
    Length: 3.968644639614441e+03
    FIPM: 5.272670350224763e-03
    EL: 2.209119555287649e+03
    EW: 1.154700538379251e+00
    EA: 9.994773028306162e-01
```

Come mostrato, tutta l'elaborazione avviene in maniera strutturata e modulare.

6. Geometrical features

In questa sezione del progetto ci siamo occupati di estrarre tutte le caratteristiche di natura geometrica. Esaminiamo dapprima il contenuto dello script "spillgeom".

```
% Script per il calcolo delle features di natura geometrica
% img: Spill oil segmentato

function [ out ] = spillgeom( img )
% 1. Calcolo del perimetro
[out.Perimeter, perim_img] = spillperim( img );

% 2. Calcolo dell'area
out.Area = spillarea( img );

% 3. Calcolo della complessità dell'oggetto
% This feature will take a small numerical value for regions with simple
% geometry and larger values for complex geometrical regions.
out.Complexity = out.Perimeter / (2 * sqrt( pi * out.Area ));

% 4. Length (L): sum of skeleton edges
% (obtained by Delaunay triangulation),
```

```

% that build the main line.

out.Length = spilldelaunay( img );

% Width (W): mean value of Delaunay triangles
% which are crossed by main line.
% out.Width = ?

% Length To Width Ratio (LWR)
% out.LWR = out.Length / out.Width;

% Compactness (Comp), defined as
% out.Comp = ( out.Length * out.Width ) / out.Area;

% 5. Calcolo del FIPM
out.FIPM = spillfipm( img );

% 6. Calcolo dei parametri basati sull fitting dell'ellisse
% Ellipse Length: value of main axe of an ellipse fitted to the data
% Ellipse Width: value of minor axe of an ellipse fitted to the data.
[ out.EL, out.EW ] = spillellipfit( img );
% Ellipse Asymetry
out.EA = 1 - ( out.EW / out.EL );

```

6.1 Calcolo del perimetro

```

% Questo script si occupa di ricavare il perimetro
% dell'oggetto in input

function [length, perim_img] = spillperim( img, debug )
% il parametro debug è opzionale
if nargin <= 1
    debug = false;
end

% L'oggetto in input deve essere binario
temp = spillbin( img );

% Produco l'immagine contenete il bordo dell'oggetto
perim_img = bwperim( temp );

% Debug grafico
if debug
    imshow(perim_img);
end

% Calcolo il perimetro
length = sum( int32( perim_img(:) ) );

```

Notiamo che in questo script è stata utilizzata la funzione `spillbin`, tale funzione si occupa di verificare che la matrice in input sia binaria, in caso negativo provvede con la binarizzazione.

```

imgbin = img;

if size(img, 3) == 3 % vuol dire che l'immagine è RGB
    level = graythresh(img);
    imgbin = im2bw(img, level);
end

```

6.2 Calcolo dell'area

Riguardo a questo calcolo, l'utilizzo di Matlab come strumento di sviluppo ci ha agevolati molto, in quanto esistono moltissime funzioni predefinite per la soluzione di problemi banali.

```
% Calcolo dell'area dell'oggetto passato in input
```

```
function [area] = spillarea( img )
```

```
% L'oggetto in input deve essere binario
```

```
temp = spillbin( img );
```

```
% Calcolo dell'area
```

```
area = bwarea( temp );
```

6.3 Triangolazione di Delaunay