

# Corso di **Sistemi Interattivi**

## **Lezione 11. Riconoscimento oggetti**

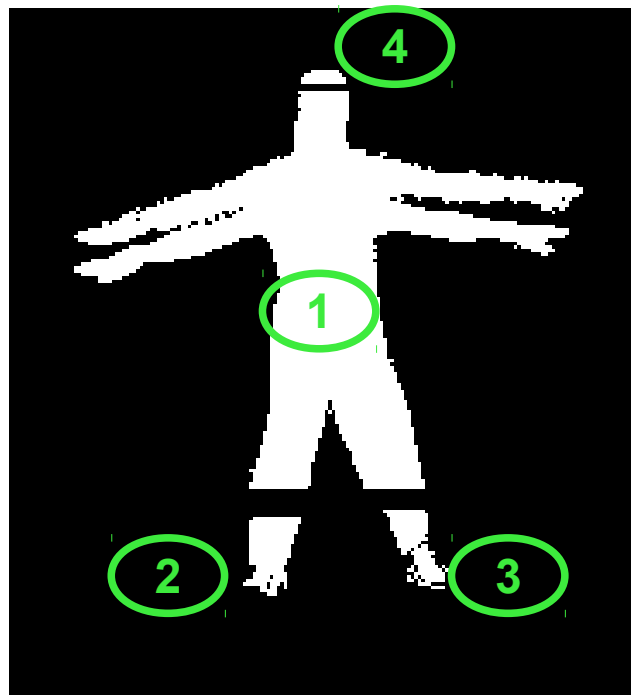
Prof. Rudy Melli ([rudymelli@ababrera.it](mailto:rudymelli@ababrera.it))

**[www.vision-e.it/si](http://www.vision-e.it/si)**

**ACCADEMIA DI BELLE ARTI DI BRERA**  
**Anno accademico 2017/2018**

# Riconoscimento di blob

- ❏ A partire da una maschera è possibile estrarre informazioni topologiche/geometriche sul suo contenuto analizzando i blob
- ❏ Un blob è un agglomerato di punti bianchi connessi (uniti) tra loro, in una maschera possono essere presenti più blob
  - ☛ Un blob è una nuvola di punti che può corrispondere ad un oggetto



In questa immagine  
ci sono 4 blob

## Libreria del corso SIPLib

- ❏ E' la libreria del corso, basata sulla libreria OpenCv for Processing, che semplifica le operazioni di segmentazione
- ❏ Per integrarla nel codice è necessario copiare il file SIPLib.pde nella cartella dello sketch
- ❏ Aggiungere una variabile di tipo SIPLib all'inizio del programma:
  - ☞ `SIPLib siplib;`
- ❏ Istanziare la variabile dentro la funzione setup con le dimensioni da elaborare a seconda del video o della camera:
  - ☞ `siplib = new SIPLib(this, 640, 480);`
  - ☞ `siplib.closingPass = 2;` ← Opzionale, serve per avere silhouette senza fori o separazioni
- ❏ Utilizzare la tecnica di segmentazione settando *SegmentationMode* od usare direttamente la funzione specifica

# Siplib - Proprietà di un blob (Topologia)

- ❏ La libreria SIPLib del corso permette di estrarre i blobs presenti nell'immagine segmentata e di accedere alle relative proprietà topologiche che li descrivono e li identificano nello spazio
- ❏ Il baricentro globale è identificato dalla variabile cog
- ❏ Solitamente si crea un ciclo per accedere ad ogni blob di tipo Contour:
  - ☞ **Contour** c = siplib.blobs.get(i);
  - ☞ **Contour** è un tipo dato delle OpenCv
- ❏ Proprietà principali:
  - ☞ Rectangle r = c.**getBoundingBox()**;  
è il rettangolo minimo che contiene il blob
  - ☞ int a = contour.area();  
area in pixel del blob
- ❏ Come baricentro si può usare il centro del rettangolo BoundingBox:
  - ☞ float cog\_x = (float)r.getCenterX();
  - ☞ float cog\_y = (float)r.getCenterY();
- ❏ Per calcolare il punto dei piedi si usa sempre il BoundingBox:
  - ☞ float foot\_x = (float)(r.x + r.width/2);
  - ☞ float foot\_y = (float)(r.y + r.height);

```
87 // Tracking multiplo
88 for(int i=0; i<siplib.blobs.size(); i++)
89 {
90     noFill();
91     color ci = siplib.GetIndexColor(i);
92     stroke(ci);
93     Contour contour = siplib.blobs.get(i);
94     Rectangle r = contour.getBoundingBox();
```

# Tecniche di segmentazione

## Utilizzare la tecnica di segmentazione scelta

- ☞ `Siplib.lightSegmentation(img, threshold, minsize);`
  - *Segmentazione in base alla luminosità per trovare le luci*
- ☞ `siplib.shadowSegmentation(img, threshold, minsize);`
  - *Segmentazione in base alla luminosità, per trovare le ombre*
- ☞ `siplib.singleDifference(img, threshold, minsize);`
- ☞ `siplib.backgroundSuppression(img, threshold, minsize, true, thMotion, minsize);`
- ☞ `siplib.backgroundSuppression(img, threshold, minsize, false, 0, 0);`
  - *Con true viene fatto anche l'aggiornamento del background automatico quando non viene rilevato movimento per 15 sec*
- ☞ `siplib.colorSegmentationPixel(img, colTrack, threshold, minsize);`
  - *Segmentazione del colore*

`img` = immagine di ingresso (webcam o video)

`threshold` = soglia (nella `colorSegmentationPixel` è la sensibilità)

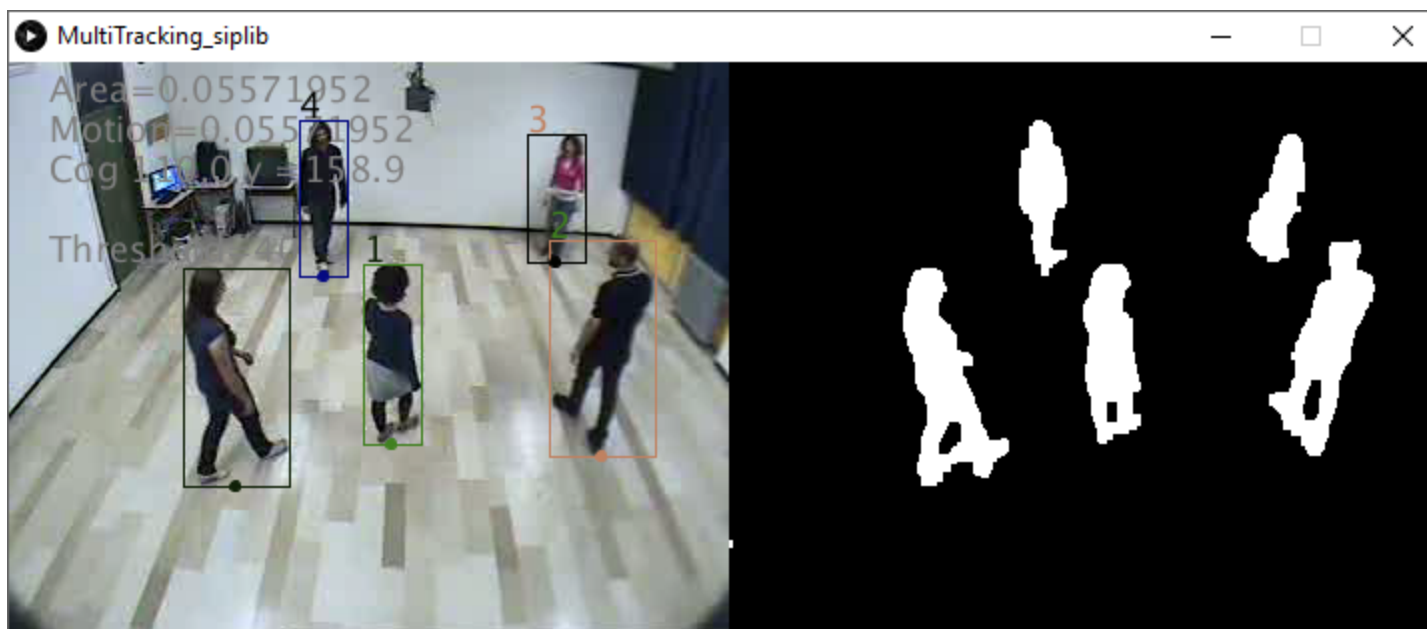
`minsize` = minima dimensione in pixel del blob più piccolo (blob più piccoli non vengono elaborati)

`thMotion` = soglia moto

`colTrack` = colore da tracciare

# MultiBlob

❏ E' possibile individuare e lavorare con più blob contemporaneamente



❏ Per sapere se c'è movimento/presenza in un rettangolo si può usare la funzione:

- ☞ `float GetRectMotion(Rectangle rc)`
  - *Utilizzo* → `float motion=siplib.GetRectMotion(rc);`
- ☞ Restituisce un valore da 0 (nessun moto/presenza) a 1 (massimo moto/presenza)

## Esercizi

- ❏ Testare le varie tecniche di segmentazione implementate in SIPLib con la webcam
- ❏ Modificare l'eventuale soglia di segmentazione collegandola al valore del mouse in modo che il valore passato abbia un range corretto (usare la funzione *map()*)
- ❏ Disegnare un cerchio nell'immagine in funzione delle coordinate del baricentro globale rilevate da SIPLib
- ❏ Testare gli esempi del corso *aa1718\_si→si→Sketches→Motion*
- ❏ Testare gli esempi in *Contributed Libraries→OpenCV For Processing*