

Corso di **Sistemi Interattivi**

Lezione 9. Riconoscimento oggetti

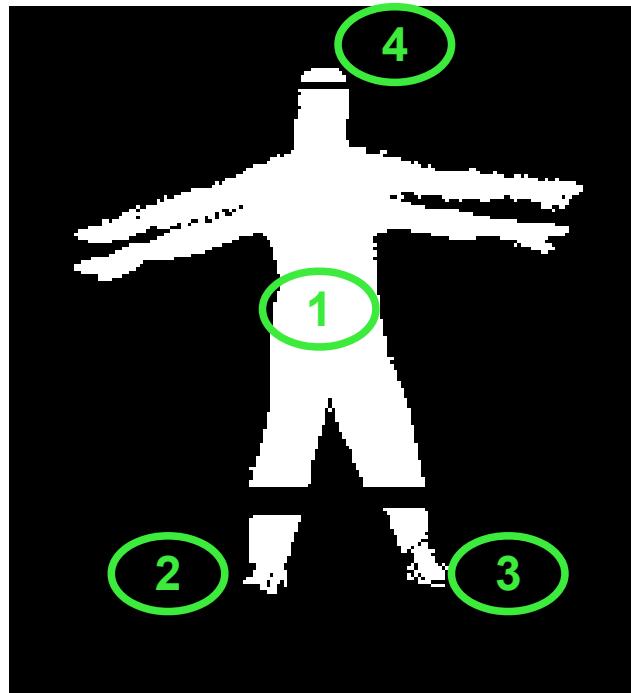
Prof. Rudy Melli (rudymelli@ababrera.it)

www.vision-e.it/si

ACCADEMIA DI BELLE ARTI DI BRERA
Anno accademico 2019/2020

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

Object detection

- ❏ I blob ci permettono di rilevare oggetti
- ❏ Un blob può essere una persona o un oggetto
- ❏ Ma una persona/oggetto può essere formata da più blob
 - ☞ “Spezzettata” durante la segmentazione
 - ☞ Si può ridurre con un’operazione morfologica chiamata *Chiusura* → parametro *ClosingPass* della SIPLib
 - ☞ Possiamo “scartare” i blob troppo piccoli e quindi poco interessanti
- ❏ Per ogni blob possono essere calcolate delle informazioni topologiche:
 - ☞ Contorno
 - ☞ Rettangolo minimo che lo contiene (chiamato anche extent o boundingbox)
 - Questo dato approssima altezza e larghezza dell’oggetto
 - ☞ Baricentro → Posizione nell’immagine
 - ☞ Numero di punti (area)
- ❏ Come riconosciamo la posizione di una persona nella stanza
 - ☞ Usiamo la posizione dei piedi
 - Indipendente dall’altezza della persona
 - Il pavimento è un piano che non varia rispetto ad una telecamera fissa

Libreria del corso SIPLib

- E' la libreria del corso, basata sulla libreria *OpenCv for Processing*, che semplifica le operazioni di segmentazione e deve essere già installata in Processing
- 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, <SegmentationMode>);`
 - ☞ `siplib.closingPass = 2;` ← Opzionale, serve per avere silhouette senza fori o separazioni
 - ☞ *SegmentationMode* è un parametro di valore 0-5 che determina la tecnica di segmentazione da utilizzare:
 - *0=ShadowSegmentation*
 - *1=SingleDifference*
 - *2=BackgroundSuppression*
 - *3=BackgroundSuppression manuale*
 - *4=ColorSegmentation*
 - *5=LightSegmentation*

Libreria del corso SIPLib

📖 Inserire all'inizio dello sketch la variabile *threshold* e *colTrack*:

- ☞ `int threshold = 50;`
- ☞ `color colTrack = 0;`

📖 Inserire, a seconda dell'uso di video o telecamera, una delle seguenti funzioni che contiene la funzione *siplib.analyze*

```
void movieEvent(Movie c) {  
    c.read();  
    siplib.analyze(c, threshold, 50, colTrack, 40, 50);  
    image_source = siplib.imgCurrent;  
}  
  
void captureEvent(Capture c) {  
    c.read();  
    siplib.analyze(c, threshold, 50, colTrack, 40, 50);  
    image_source = siplib.imgCurrent;  
}
```

📖 *siplib.analyze* ha 6 parametri:

- ☞ Immagine
- ☞ Soglia segmentazione
- ☞ Dimensione minima blob (pixel)
- ☞ Colore da tracciare (valido solo per colorSegmentation)
- ☞ Soglia movimento (Valido solo per background suppression smart)
- ☞ Dimensione minima blob (Valido solo per background suppression smart)

Tecniche di segmentazione

 Utilizzare la tecnica di segmentazione scelta

- ☞ 0 = **shadowSegmentation**
 - *Segmentazione in base alla luminosità, **per trovare le ombre***
- ☞ 1 = **singleDifference**
- ☞ 2 = **backgroundSuppression**
 - Il quarto parametro, con *true*, imposta una modalità automatica di aggiornamento del background quando non viene rilevato movimento per 15 sec
- ☞ 3 = **backgroundSuppression**
 - *Background suppression con memorizzazione manuale del background, cioè il background viene memorizzato quando viene*
- ☞ 4 = **colorSegmentation**
 - ***Segmentazione del colore***
- ☞ 5 = **lightSegmentation**
 - ***Segmentazione in base alla luminosità, per trovare le luci***

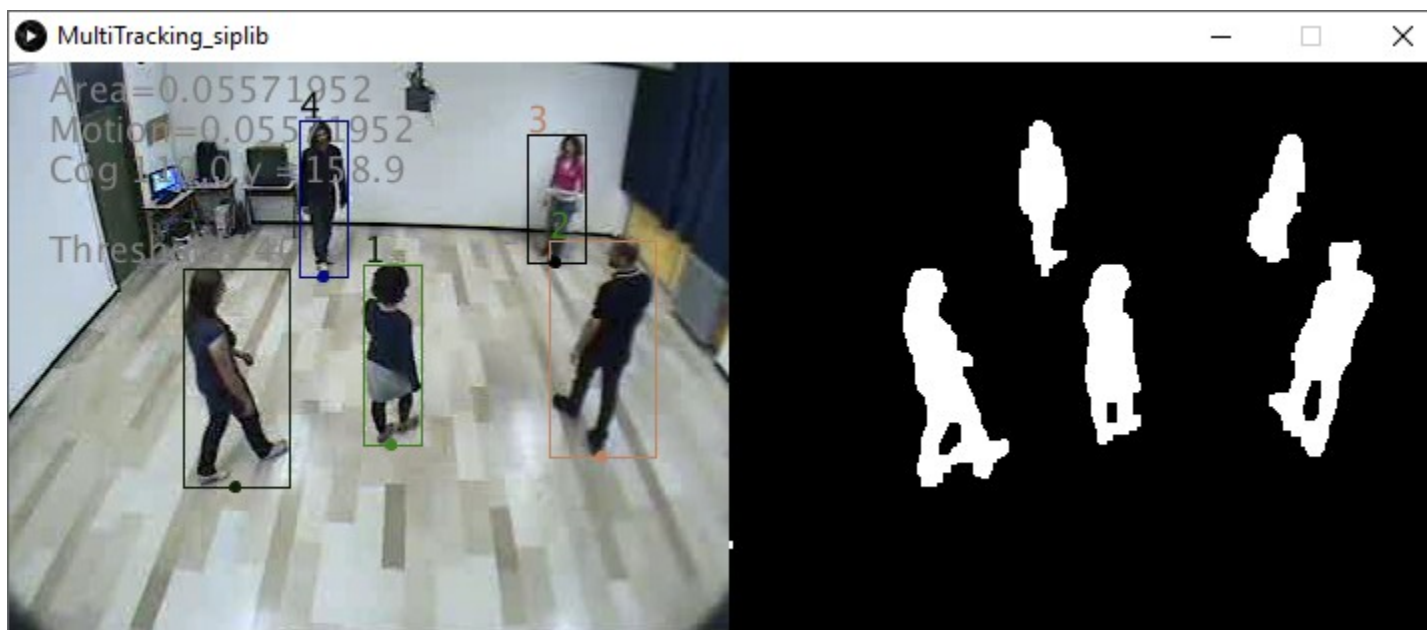
Siplib - Proprietà di un blob (Topologia)

- ❏ 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();
```

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 *Sketches→Motion*
- Testare gli esempi in *Contributed Libraries→OpenCV For Processing*
- Testare gli esempi in *Contributed Libraries→BoofCv For Processing*