

Prova pratica

Primo tema:

1. Il candidato implementi un algoritmo di edge detection a sua scelta partendo da zero (cioè senza usare algoritmi per l'edge detection già implementati in librerie di computer vision) e elabori l'immagine fornita per estrarre i bordi presenti nell'immagine. L'immagine ottenuta dall'edge detection deve essere salvata nella cartella messa a disposizione (punti 13).
2. Il wavefront propagation algorithm è un algoritmo per il motion planning di robot in ambienti la cui mappa è nota ed è rappresentata con una occupancy grid. Il valore di ogni elemento della occupancy grid sia: '-1' se l'elemento è occupato da un ostacolo, oppure un numero naturale che rappresenta il numero di mosse necessario a raggiungere il GOAL (rappresentato dal valore 0) a partire dalla posizione dell'elemento.

Dato il file di testo fornito < occupancy grid map.txt> che rappresenta l'occupancy grid di un ambiente. Il candidato implementi il wavefront propagation algorithm in un linguaggio di programmazione a sua scelta. L'output deve essere un file di testo simile a quello rappresentato in figura (cioè una matrice di numeri, ma in modalità testo, non in modalità grafica). L'output deve essere salvato nella cartella messa a disposizione (punti 17).

Inoltre, per entrambi gli esercizi il candidato illustri brevemente il procedimento seguito sui fogli protocollo messi a sua disposizione.

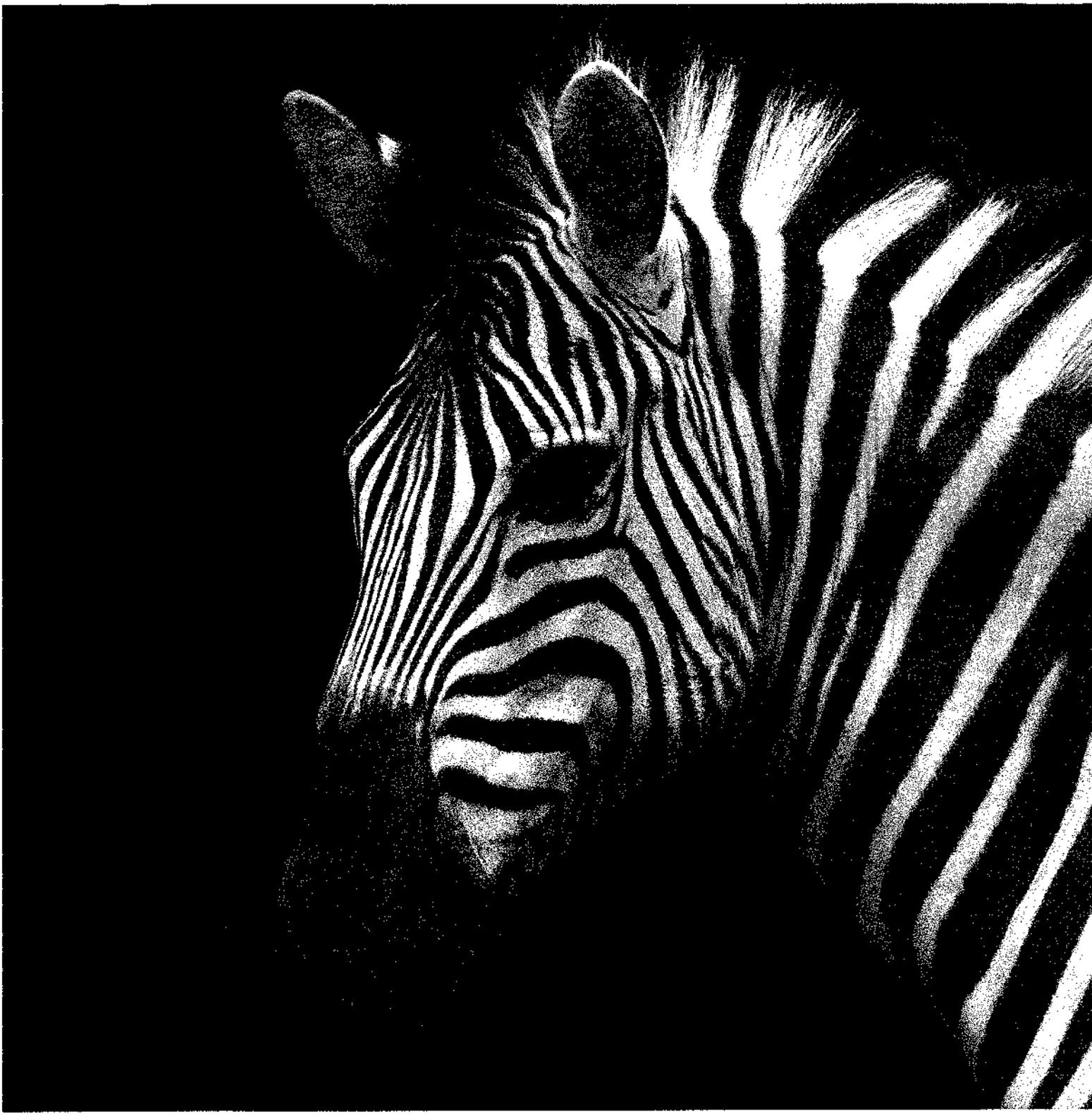
WAVEFRONT PROPAGATION ALGORITHM

1. Initialize $W_0 = X_G; i = 0.$
2. Initialize $W_{i+1} = \emptyset.$
3. For every $x \in W_i$, assign $\varphi(x) = i$ and insert all unexplored neighbors of x into $W_{i+1}.$
4. If $W_{i+1} = \emptyset$, then terminate; otherwise, let $i := i + 1$ and go to Step 2.

Figure 1. Pseudocodice del wavefront algorithm.

22	21	22	21	20	19	18	17	16	17	
21	20							15	16	
20	19							14	15	
19	18	17	16	15	14	13	12	13	14	
18	17	16	15	14	13	12	11	12	13	
								10	11	12
								9	10	11
3	2	1	2	3				8	9	10
2	1	0	1	2				7	8	9
3	2	1	2	3	4	5	6	7	8	

Figure 2. Esempio di output del wavefront algorithm

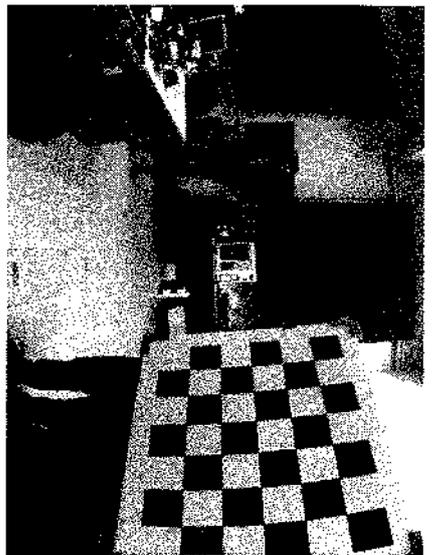
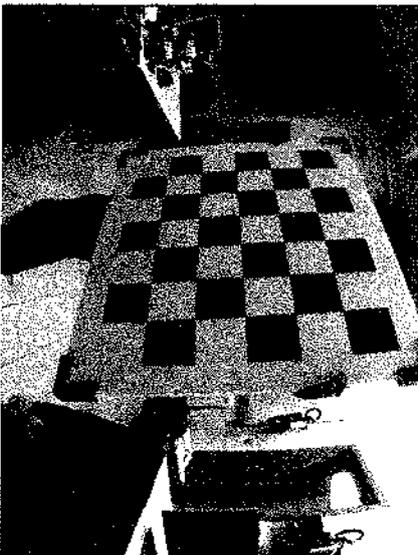
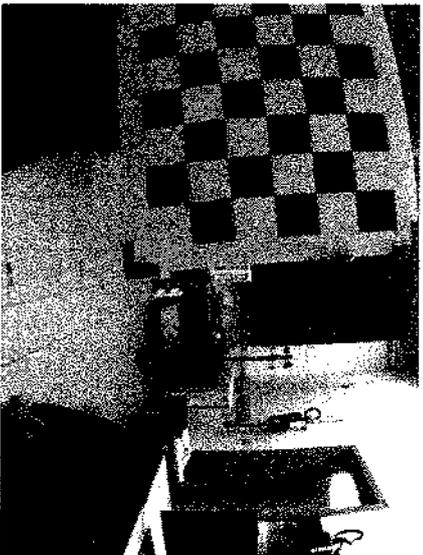
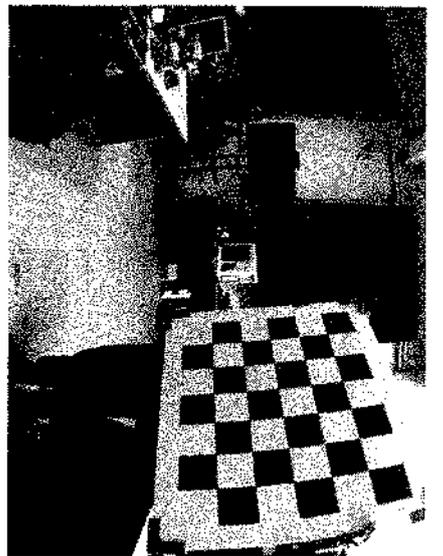
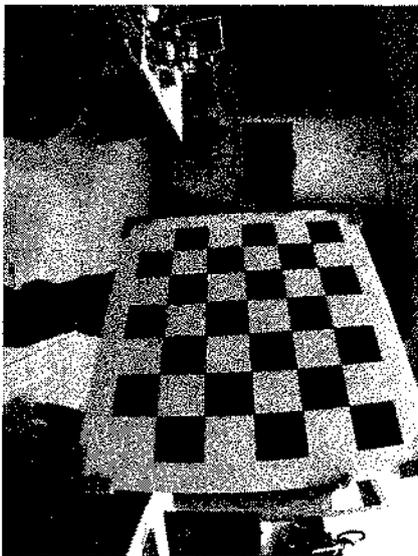
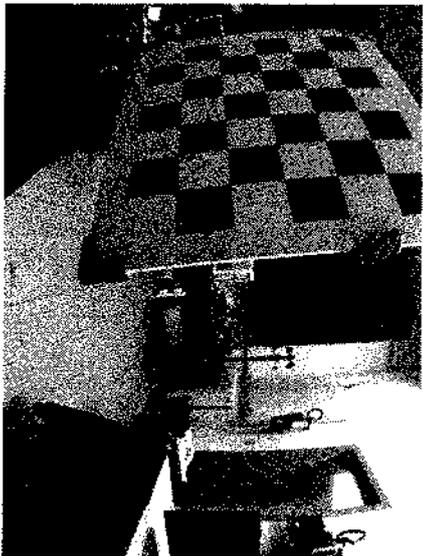
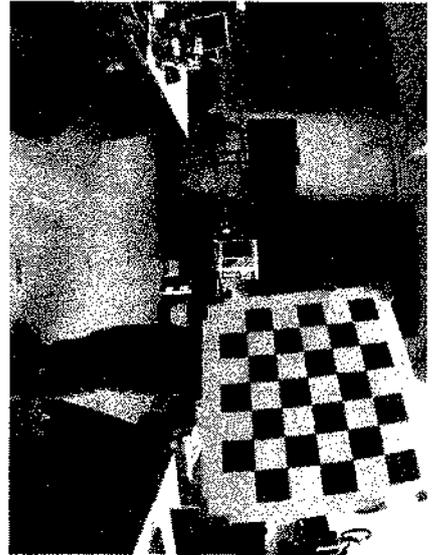
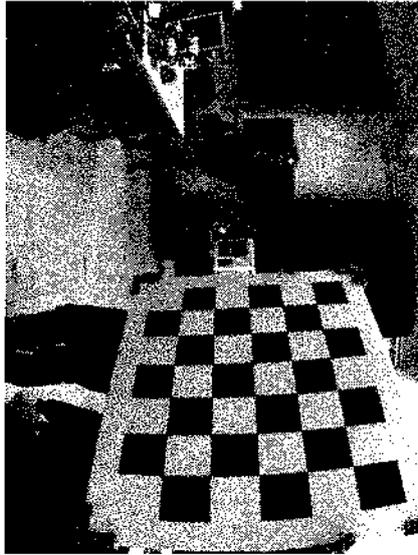
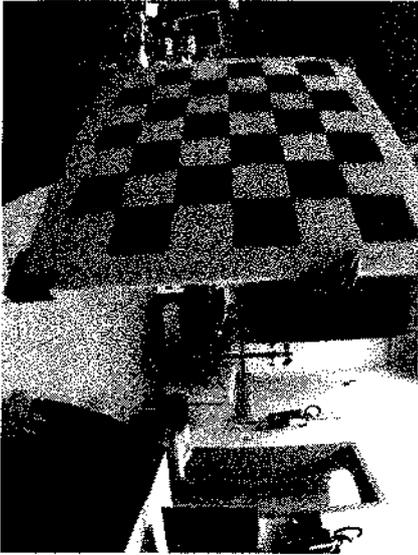


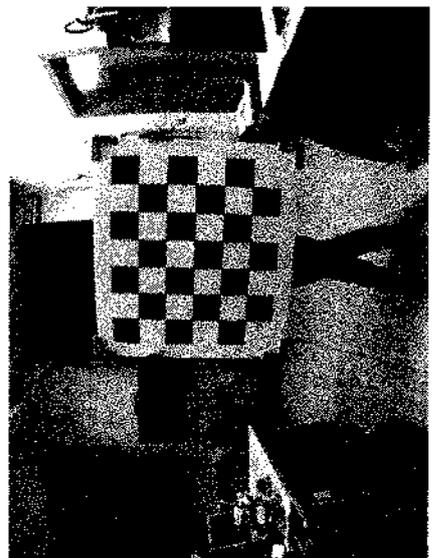
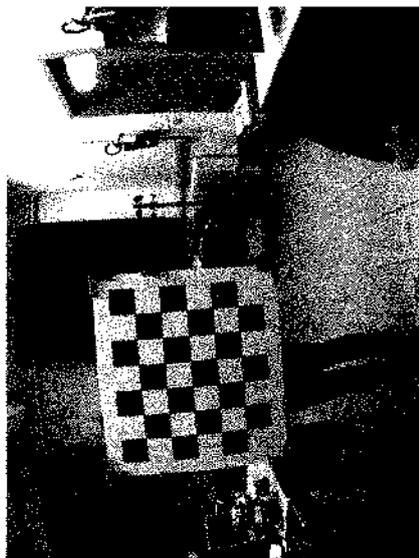
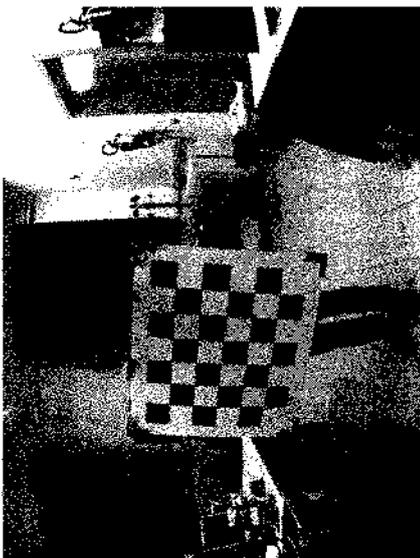
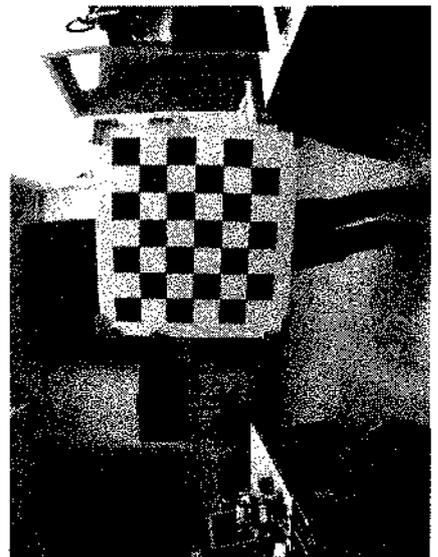
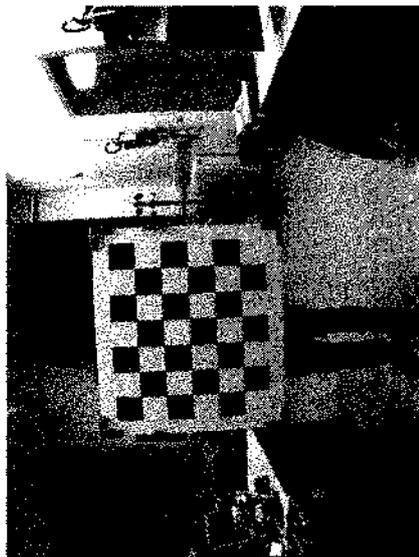
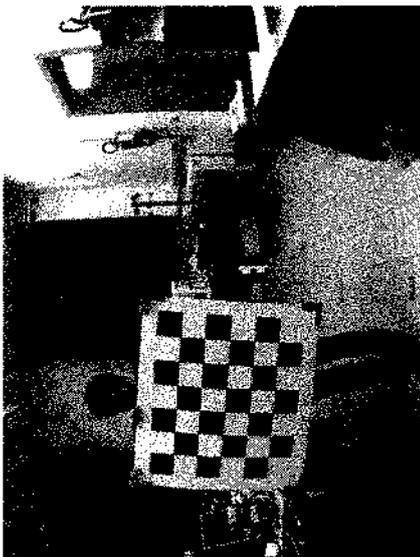
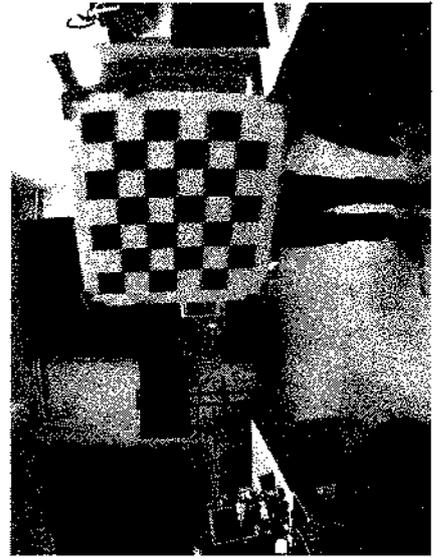
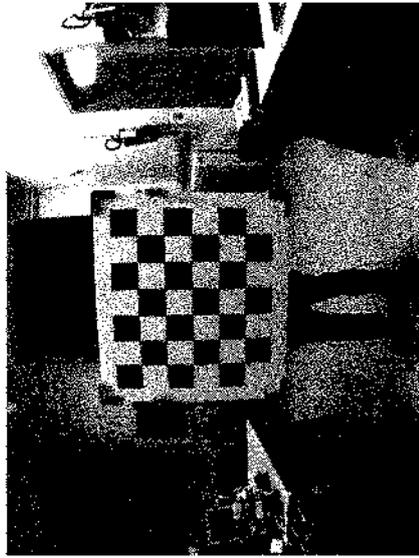
Prova pratica

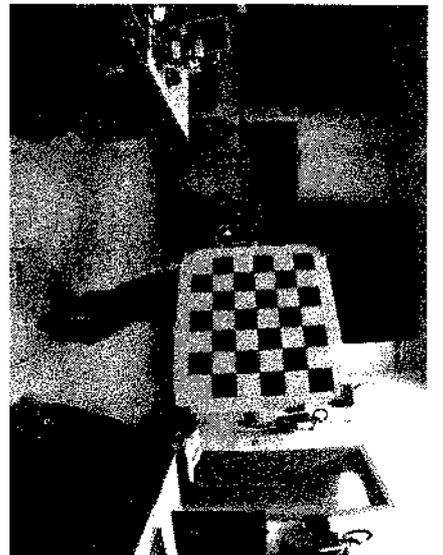
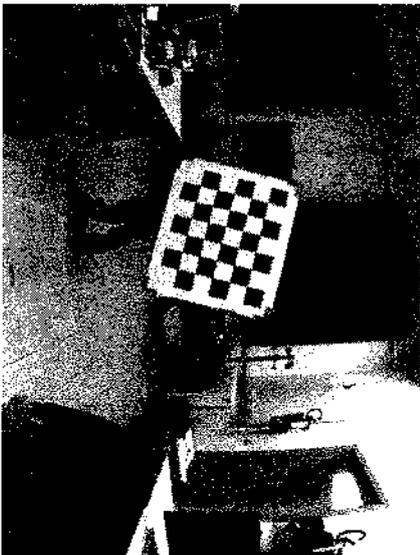
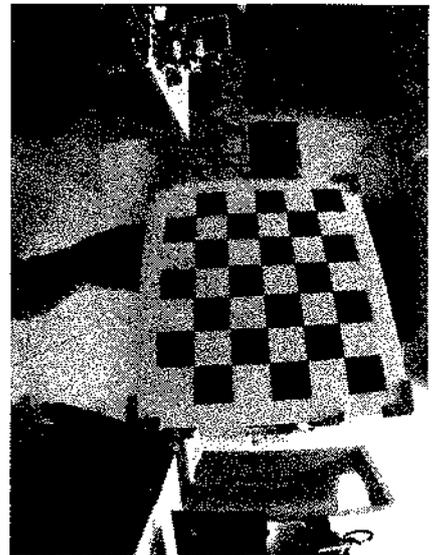
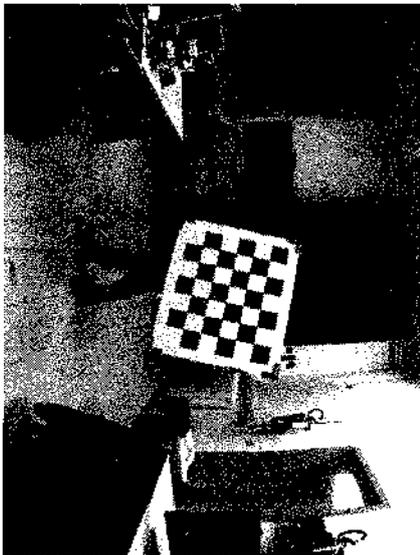
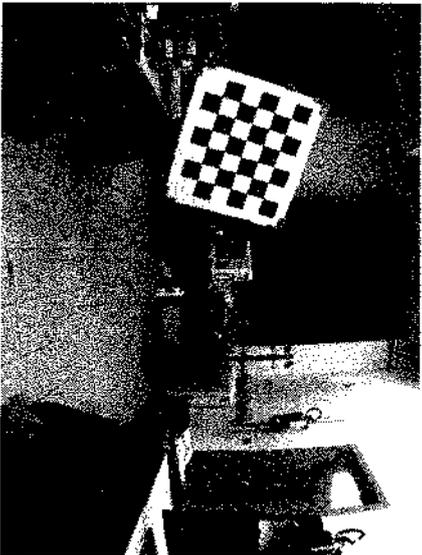
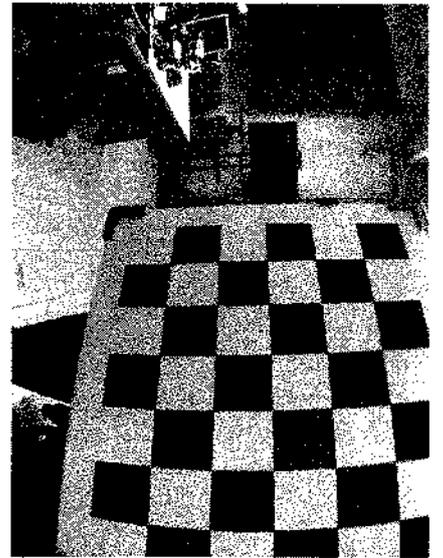
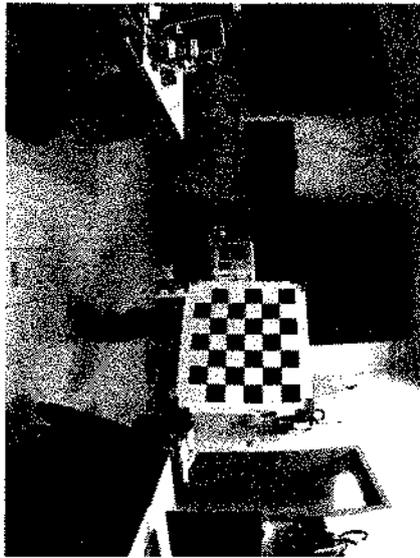
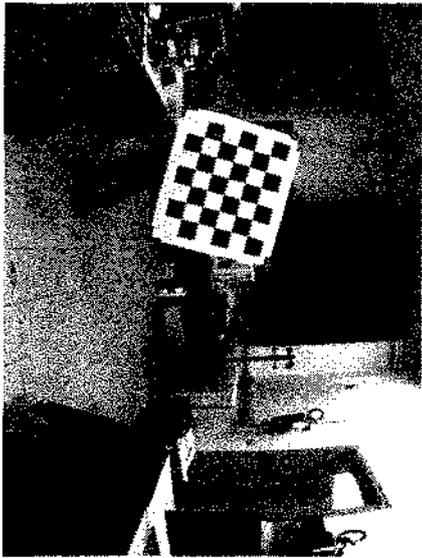
Secondo tema:

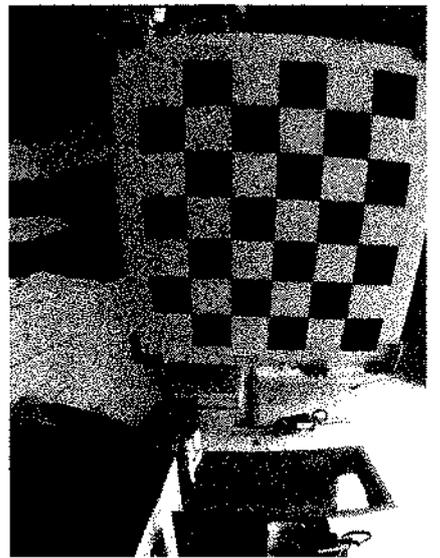
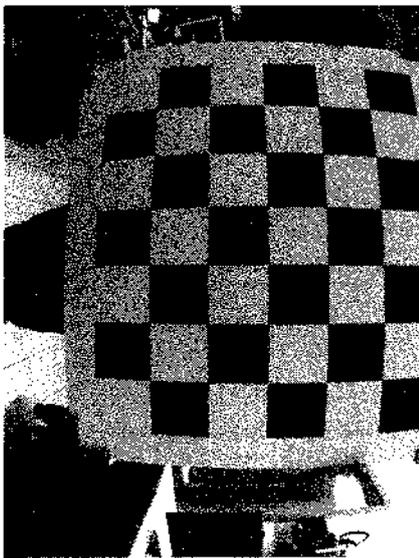
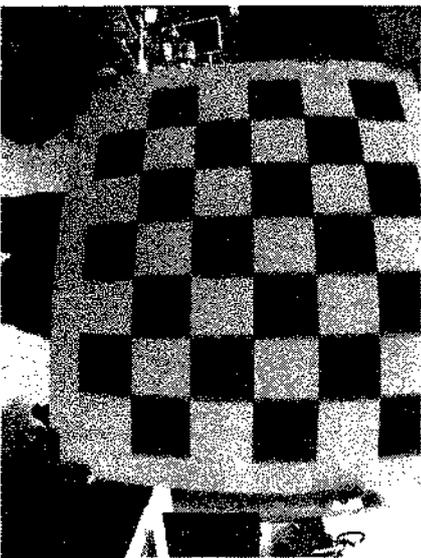
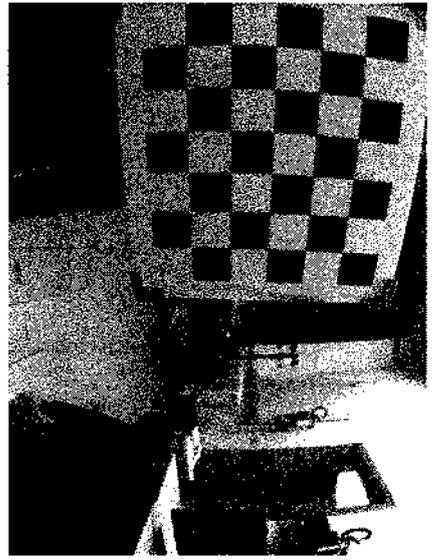
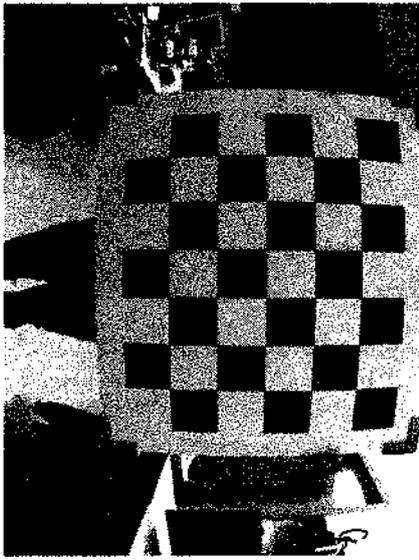
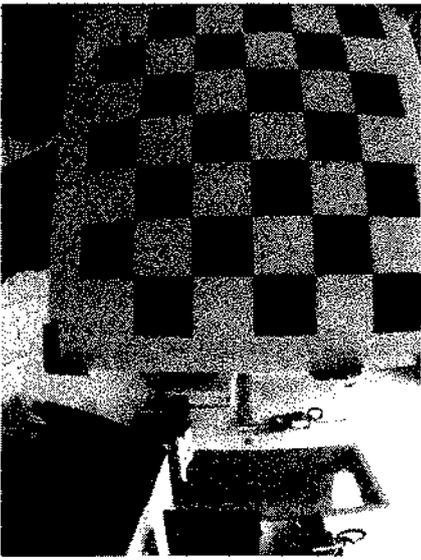
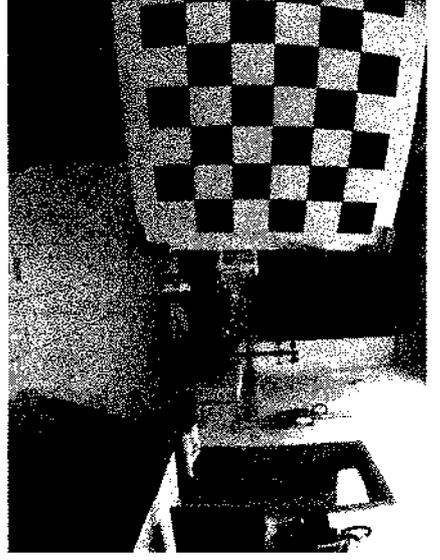
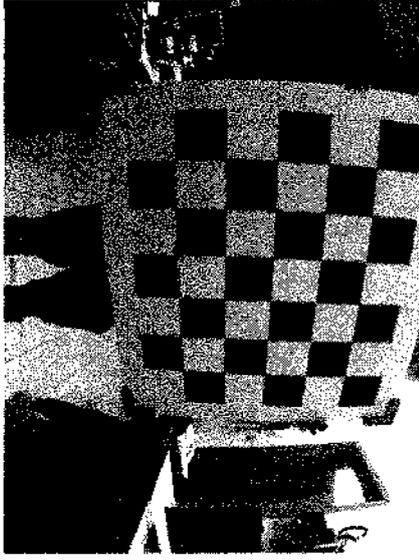
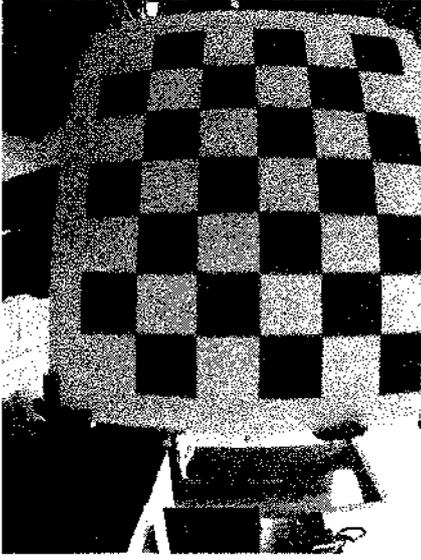
1. Il candidato implementi un algoritmo di segmentazione di colore sull'immagine fornita. Il colore deve essere selezionabile dall'utente con un metodo a scelta dal candidato (ad es. mouse o soglie impostabili in modo interattivo dall'utente, ecc.). L'immagine ottenuta dalla segmentazione del colore e uno screenshot dell'interfaccia di selezione del colore devono essere salvati nella cartella messa a disposizione **(punti 15)**.
2. Utilizzando il dataset di immagini di una scacchiera fornito, il candidato calcoli i parametri di calibrazione della telecamera che ha acquisito le immagini sviluppando un programma che utilizzi le librerie software disponibili sul computer a disposizione (Matlab o OpenCV). Il candidato salvi i parametri di calibrazione così calcolati su un file di testo assieme alle immagini della scacchiera de-distorte nella cartella messa a disposizione **(punti 15)**.

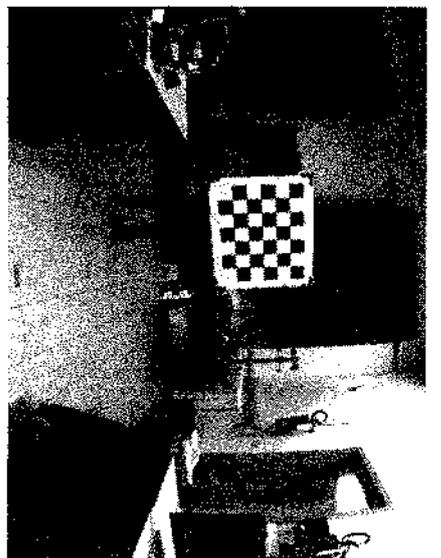
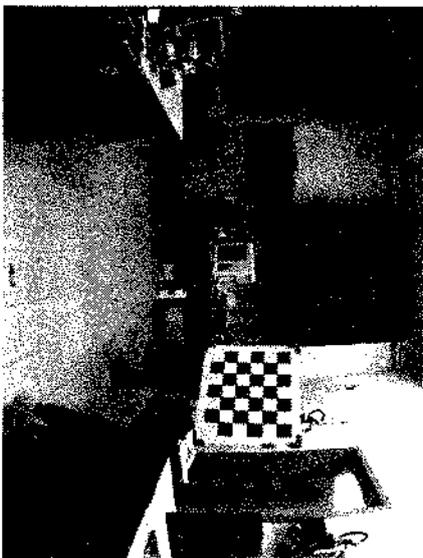
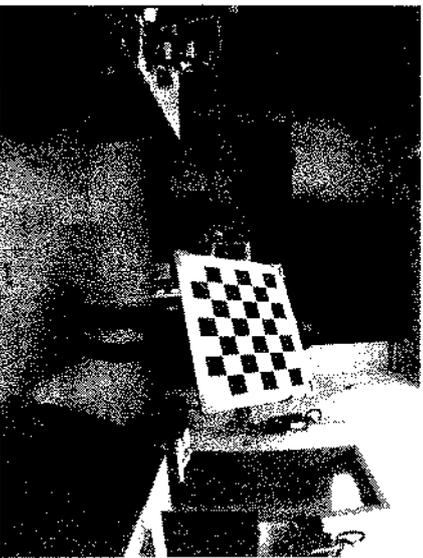
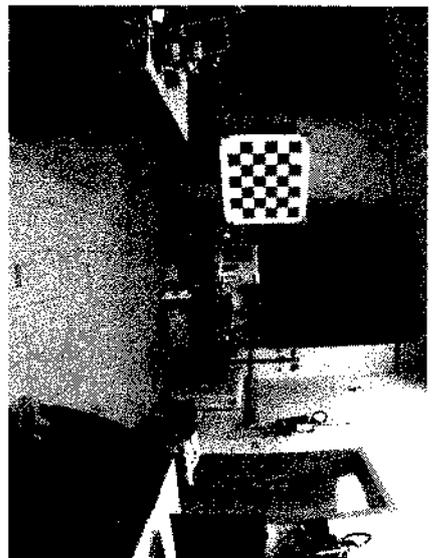
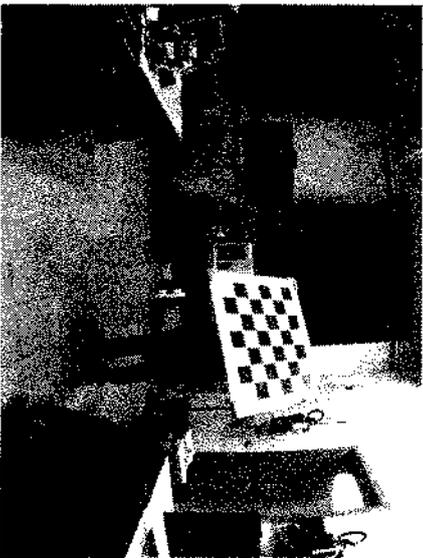
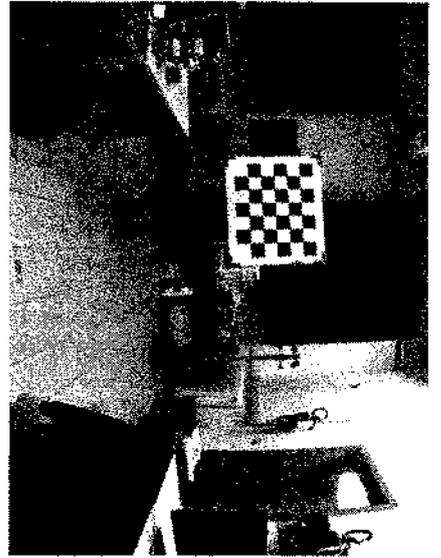
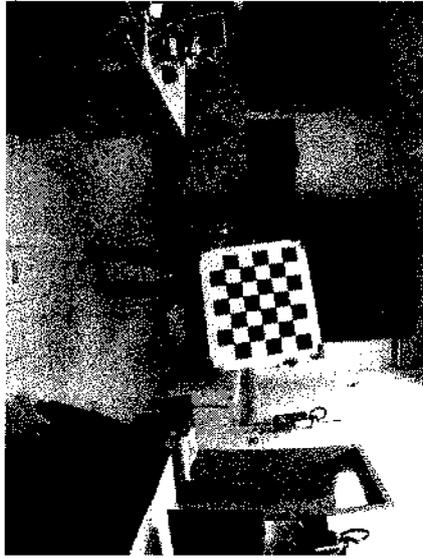
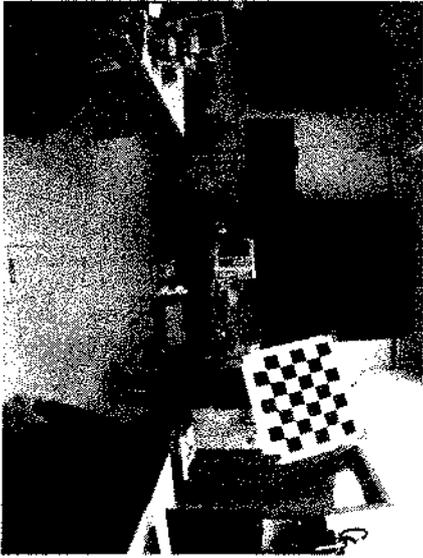
Inoltre, per entrambi gli esercizi il candidato illustri brevemente il procedimento seguito sui fogli protocollo messi a sua disposizione.

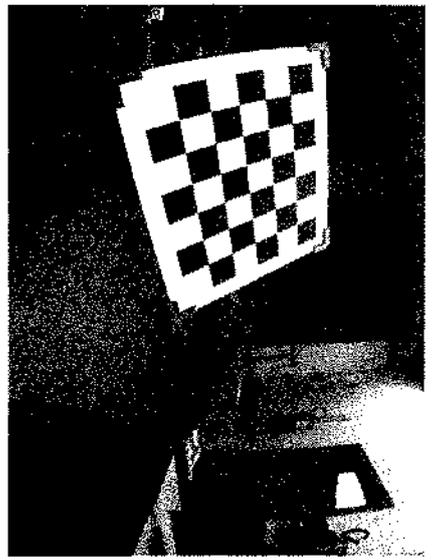
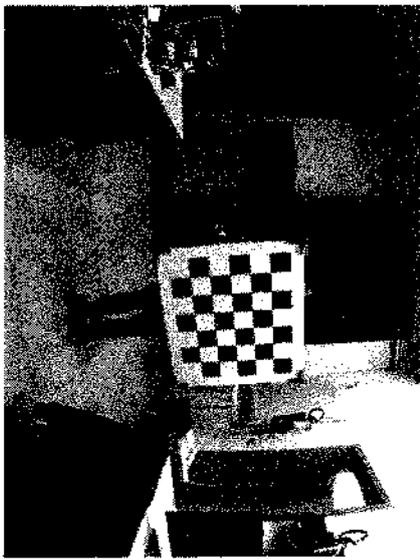
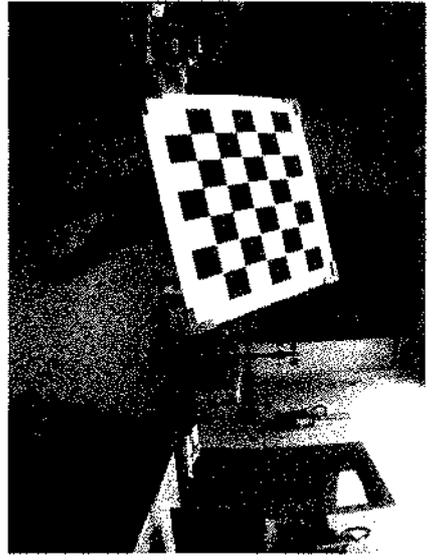
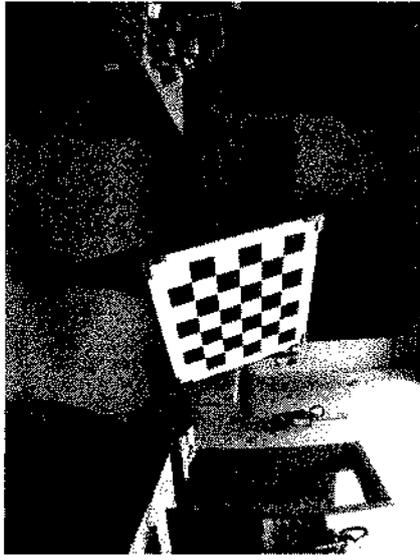
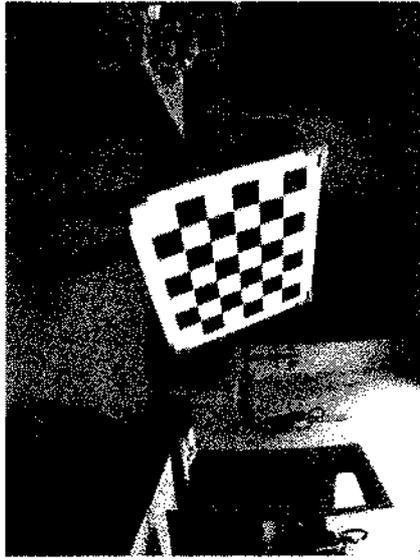
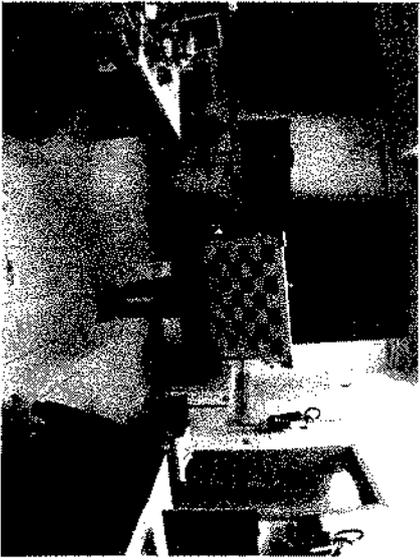


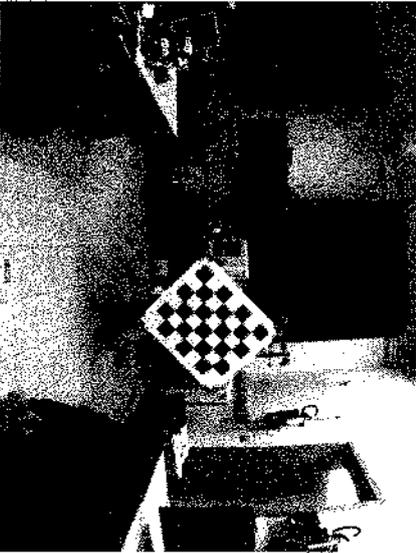
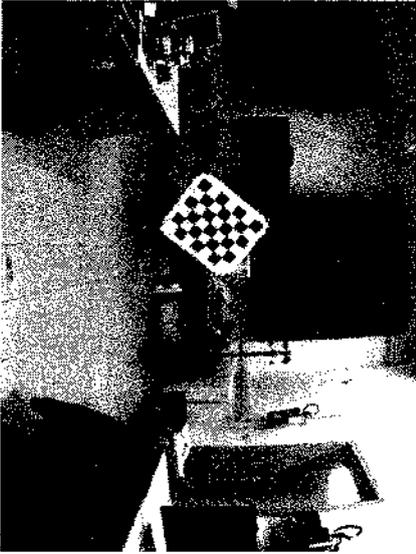


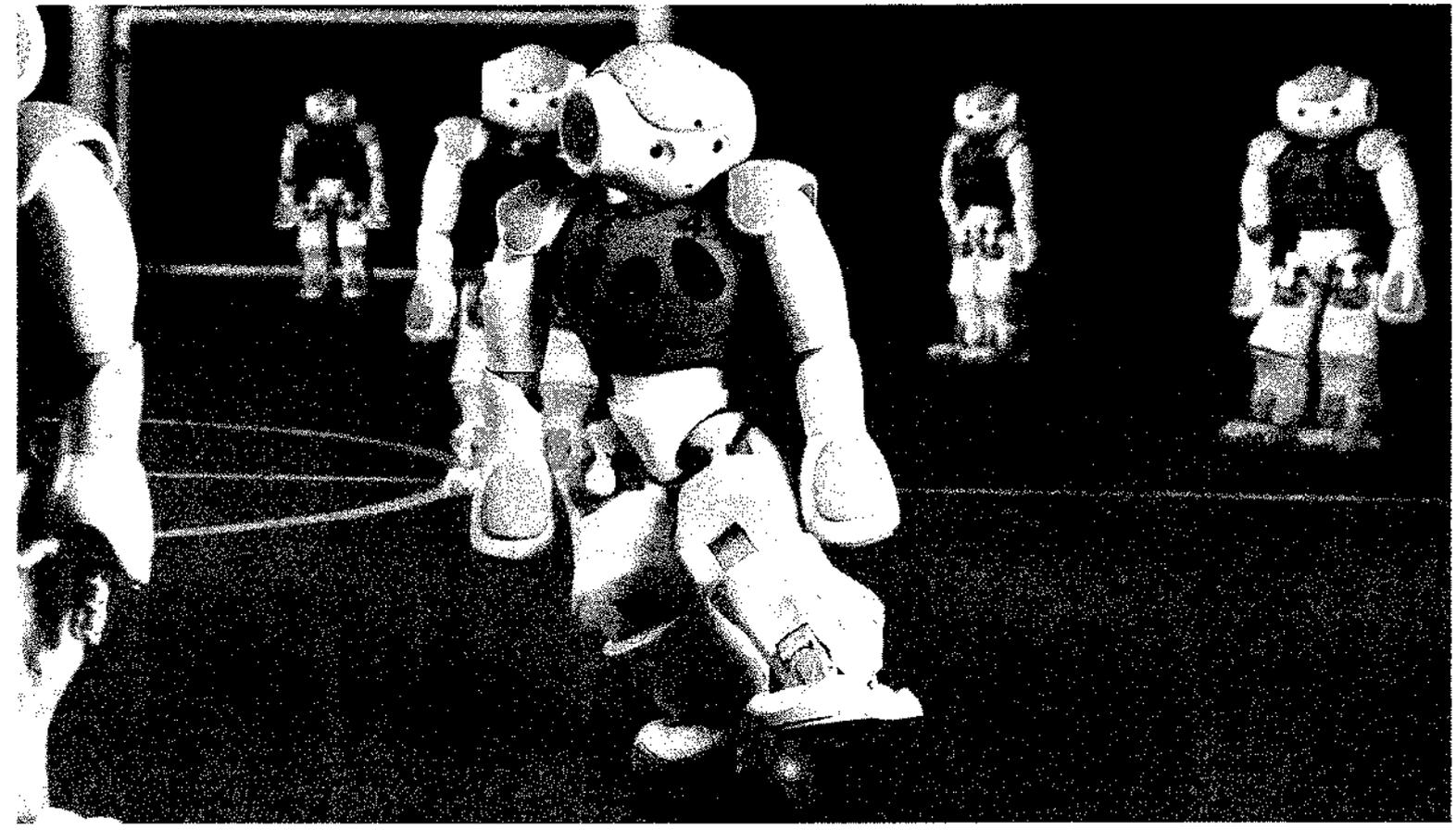












Prova pratica

Terzo tema:

1. Il candidato implementi un algoritmo di smoothing a sua scelta partendo da zero (cioè senza usare algoritmi di filtraggio già implementati in librerie di computer vision) e elabori l'immagine fornita per generare una immagine filtrata. L'immagine ottenuta con lo smoothing deve essere salvata nella cartella messa a disposizione (punti 13).
2. Il wavefront propagation algorithm è un algoritmo per il motion planning di robot in ambienti la cui mappa è nota ed è rappresentata con una occupancy grid. Il valore di ogni elemento della occupancy grid sia: '255' se l'elemento è occupato da un ostacolo, oppure un numero naturale che rappresenta il numero di mosse necessario a raggiungere il GOAL (rappresentato dal valore 0) a partire dalla posizione dell'elemento.

Dato il file di testo fornito <occupancy grid map.txt> che rappresenta l'occupancy grid di un ambiente. Il candidato implementi il wavefront propagation algorithm in un linguaggio di programmazione a sua scelta. L'output deve essere un file di testo simile a quello rappresentato in figura (cioè una matrice di numeri, ma in modalità testo, non in modalità grafica). L'output deve essere salvato nella cartella messa a disposizione (punti 17).

Inoltre, per entrambi gli esercizi il candidato illustri brevemente il procedimento seguito sui fogli protocollo messi a sua disposizione.

WAVEFRONT PROPAGATION ALGORITHM

1. Initialize $W_0 = X_G; i = 0$.
2. Initialize $W_{i+1} = \emptyset$.
3. For every $x \in W_i$, assign $\phi(x) = i$ and insert all unexplored neighbors of x into W_{i+1} .
4. If $W_{i+1} = \emptyset$, then terminate; otherwise, let $i := i + 1$ and go to Step 2.

Figure 3. Pseudocodice del wavefront algorithm.

22	21	22	21	20	19	18	17	16	17	
21	20							15	16	
20	19							14	15	
19	18	17	16	15	14	13	12	13	14	
18	17	16	15	14	13	12	11	12	13	
								10	11	12
								9	10	11
3	2	1	2	3				8	9	10
2	1	0	1	2				7	8	9
3	2	1	2	3	4	5	6	7	8	

Figure 4. Esempio di output del wavefront algorithm

