Università di Parma

Corso di Laurea Magistrale in Ingegneria Informatica

Fondamenti di Visione Artificiale

a.a. 2019/20

PROVA PRATICA 07-01-2020

NOME:

COGNOME:

MATRICOLA:

WORKSTATION N°:

Non è consentito scambiarsi materiale via rete (ovviamente).

Per accedere ai pixel è consentito l'uso di funzioni OpenCv di alto livello come `at<>()`.

Salvare l'esame in un file COGNOME_MATRICOLA.zip.

FIRMA

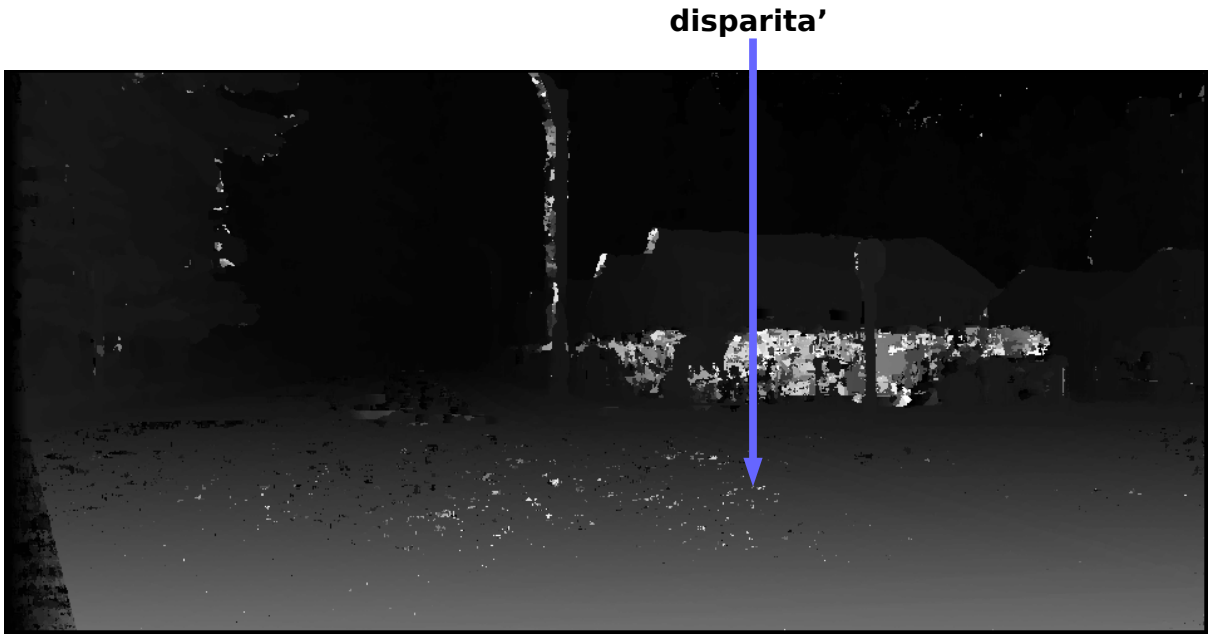
ES1: STEREO MATCHING

Data una coppia di immagini sinistra-destra, **rettificate**, L.pgm e R.pgm, calcolare la corrispondente immagine di **disparita'** con la metrica **Sum of Absolute Distances (SAD)**.

Utilizzare la funzione protitipo fornita:

```
void mySAD_Disparity7x7(const cv::Mat & left_image, const cv::Mat & right_image, cv::Mat & out)
```

Esempio:



Dettagli:

1. Confrontare il vicinato 7*x7* di ogni pixel dell’immagine **sinistra** (r,c) con il corrispondente vicinato 7*x7* dei *possibili candidati sull’immagine* ***destra***. Il confronto va fatto tramite la somma delle differenze in valore assoluto.
2. Il candidato sull’immagine destra con *somma delle differenze minore* e’ il corrispondente cercato.
3. Salvare il valore di disparita’ *d* nell’immagine di *out* come un CV_32FC1 in posizione (r,c).
4. Limitare il range di ricerca a 128 colonne → **disparita nel range [0,127]**
5. NON e’ un problema risolvibile con feature matching e key points. Il confronto va fatto pixel per pixel (stereo matching).
6. Esecuzione del codice:

```
./simple ../images/L.pgm ../images/R.pgm
```

HINTS

1. su quale riga dell’immagine destra devo cercare il corrispondente del pixel di destra? Le immagini sono **rettificate, quindi i piani immagine sono paralleli e complanari**.
2. una volta nota la riga, quali colonne ha senso espolorare? Quelle di destra o quelle di sinistra?
3. si puo’ fare con 5 cicli innestati

ES2: VDISPARITY

Creare una nuova immagine `vdisp` di altezza pari *all'altezza dell'immagine di input* e *larghezza pari a 128*, in cui ogni riga rappresenti un **istogramma dell'immagine disparita' *della corrispondente riga***. Consideriamo solo le disparita' **maggiori di zero**.

Utilizzare la funzione prototipo fornita:

```
void VDisparity(const cv::Mat & disp, cv::Mat & out)
```

HINTS

1. Notare che 128, le colonne di `vdisp`, sono esattamente tutti i possibili valori di disparita' [0-127].
2. Ogni pixel della riga r-esima dipende solo dai valori della corrispondente riga r-esima dell'immagine di disparita'
3. In pratica ogni pixel contiene il conteggio di quante disparita' ci sono sulla riga r-esima di valore `floor(disp) == colonna`