

Università di Parma
Corso di Laurea Magistrale in Ingegneria Informatica
Fondamenti di Visione Artificiale
a.a. 2019/20

PROVA PRATICA 28-01-2020

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<>()`.

Esecuzione del codice:

```
./simple ../images/left.pgm ../images/dsi/bin
```

Salvare l'esame in un file COGNOME_MATRICOLA.zip.

FIRMA

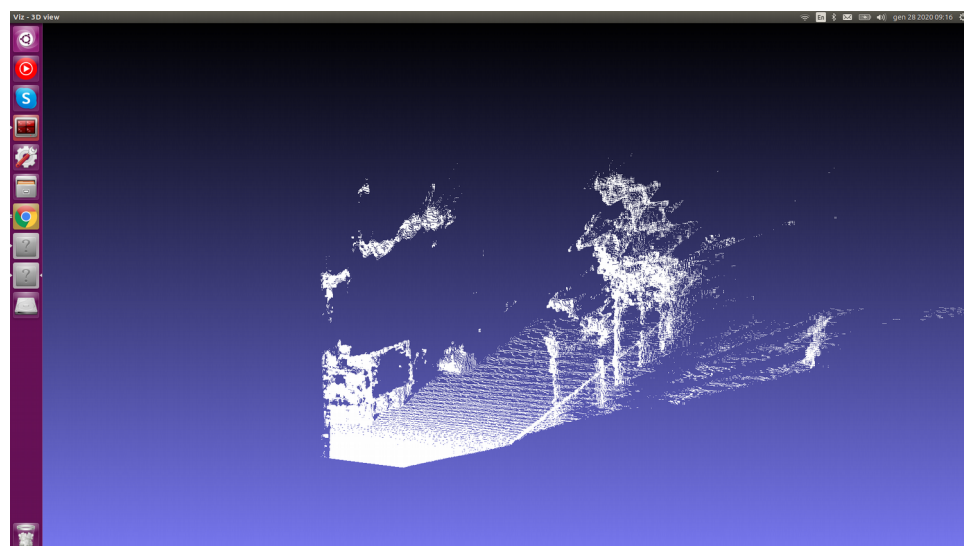
ES1: CALCOLO POINT CLOUD

Viene fornito un file “dsi.bin” che contiene le disparita’ gia’ precalcolate di una coppia destra sinistra:



Utilizzando le apposite formule, calcolare i punti **3D** corrispondenti ad ogni valore di (riga, colonna, disparita’). Le immagini sono di dimensione 960x480, i parametri di calibrazione (focale, centro ottico e baseline) sono gia’ forniti nel codice.

Il risultato dovrebbe apparire cosi’:



NOTA: per l’esame non e’ necessario visualizzare la point cloud con Viz; se non lo avete installato nel vostro OpenCV, proseguite con il resto dell’esame.

ES2: PLANE FITTING

Data la point cloud calcolata al punto precedente (quindi non la disparita', ma i punti 3D), utilizzare il metodo **RANSAC** per calcolare il **piano che meglio approssima i punti**.

Una volta ottenuti gli inlier del modello migliore, creare un'immagine che contiene solamente i pixel corrispondenti. Dovreste ottenere un'immagine in cui e' presente il piano stradale e poco piu', essendo esso il piano prevalente nell'immagine.



HINTS:

1. Attenzione che viene richiesto di applicare il **metodo** RANSAC ad un particolare problema (plane fitting), diverso dal calcolo dell'omografia come visto nell'assegnamento. Il **metodo** e' lo stesso, ma il modello che vogliamo calcolare no.
2. Un piano in 3D puo' essere calcolato a partire 3 punti. Vi viene gia' fornita la funzione opportuna `void plane3points(cv::Point3f p1, cv::Point3f p2, cv::Point3f p3, float & a, float & b, float & c, float & d)`
3. La distanza tra un punto 3D ed un piano puo' essere calcolata con la funzione fornita `float distance_plane_point(cv::Point3f p, float a, float b, float c, float d)`
4. Si tratta quindi di selezione campioni di 3 punti, calcolare il piano, calcolare gli inliers, ecc. ecc.
5. Per ogni inlier ricordatevi di salvare anche la corrispondente riga-colonna, per poter poi disegnare l'immagine