



UNIVERSITÀ DI PARMA

Laboratory Model Fitting

- Vengono forniti
 - una immagine
 - alcuni file che contengono, riga per riga:
 - le coordinate di punti dell'immagine stessa.
- Sviluppare, a partire dallo scheletro fornito, il codice che opera il fitting di una retta utilizzando un approccio RANSAC

- Si legga il file di punti passato per argomento in un `std::vector<cv::Point>`
 - Si rammenta che ogni riga contiene le coordinate di un punto

- In accordo a quanto visto si stimi il numero di iterazioni, tenendo conto che il nome del file contiene la percentuale (approssimativa) degli outliers presenti nel caso in cui il δ utilizzato sia pari a 30 pixel

- Tenendo conto del numero di iterazioni calcolato al punto precedente, si sviluppi il codice che realizza il fitting di una retta. Si disegni sulla immagine di output la retta (`cv::line()`), gli inlier e gli outlier utilizzando colori differenti.



- Si modifichi il ciclo precedente facendolo terminare quando si raggiunge (quasi) la percentuale di inlier o outlier previsti. Si stampi il numero di iterazioni calcolate al punto (1) e quelle effettivamente operate.
- A tal fine si ipotizzi che la distribuzione dei punti nella fascia 30 attorno alla retta non abbia andamento gaussiano ma “bianco”



- Vengono forniti
 - una immagine
 - alcuni file che contengono, riga per riga:
 - le coordinate di punti dell'immagine stessa.
- Sviluppare, a partire dallo scheletro fornito, il codice che opera il fitting di una retta utilizzando un approccio RANSAC



- Si legga il file di punti passato per argomento in un `std::vector<cv::Point>`
 - Si rammenta che ogni riga contiene le coordinate di un punto



- In accordo a quanto visto si stimi il numero di iterazioni, tenendo conto che il nome del file contiene la percentuale (approssimativa) degli outliers presenti nel caso in cui il δ utilizzato sia pari a 30 pixel



- Tenendo conto del numero di iterazioni calcolato al punto precedente, si sviluppi il codice che realizza il fitting di una retta. Si disegni sulla immagine di output la retta (`cv::line()`), gli inlier e gli outlier utilizzando colori differenti.





- Si modifichi il ciclo precedente facendolo terminare quando si raggiunge (quasi) la percentuale di inlier o outlier previsti. Si stampi il numero di iterazioni calcolate al punto (1) e quelle effettivamente operate.
- A tal fine si ipotizzi che la distribuzione dei punti nella fascia 30 attorno alla retta non abbia andamento gaussiano ma “bianco”