Progetto Matlab

Depth Map per ridurre falsi positivi in Face Detection

Mattia Maglie 1189315 Francesco Marcato 1189319
Marco Martini 1189321

Descrizione Metodo

Descrizione Testuale

Data la matrice DepthDATA, considero per ogni elemento il campo (di indice 2) contenente la matrice di valori di profondità.

Dalla matrice viene poi presa la riga centrale dalla quale vengono tolti i valori a 0 (che indicano un errore del sensore) e su cui vi si effettua una "regressione parabolica". Per la regressione parabolica si usa la funzione fit del Curve Fitting Toolbox.

Con i coefficienti del curve fit vengono applicate insieme 3 tecniche diverse:

1. **PARABOLIC EXISTENCE**
   Controllo che il risultato del fit sia una parabola. Nel caso in cui non sia una parabola infatti, il calcolo del vertice (-b/2a) restituisce NaN, in quanto il coefficiente di 2° grado è 0 e si ha una divisione per 0. Se non è una parabola contrassegno direttamente l'elemento come *NonFace*. Questa tecnica a sé ha una Precision del 100%.
2. **CONCAVITY CHECK**
   Nel caso in cui sia una parabola, controllo che questa sia concava. Nel caso di parabola concava ho il coefficiente a > 0. Quindi se il coefficiente a è minore o uguale a 0 contrassegno l'elemento come *NonFace*.
   Precision circa del 94,6%.
3. **VERTEX POSITION**
   Se i primi due controlli vengono superati, avendo quindi una parabola concava, controllo che il vertice di questa sia all'interno di un range specifico:

   $A = matrixHCenter - marginRate \cdot matrixWidth$

   $B = matrixHCenter + marginRate \cdot matrixWidth$

   Dove $matrixHCenter$ è la posizione della colonna centrale della matrice di profondità e $matrixWidth$ è la larghezza della matrice.

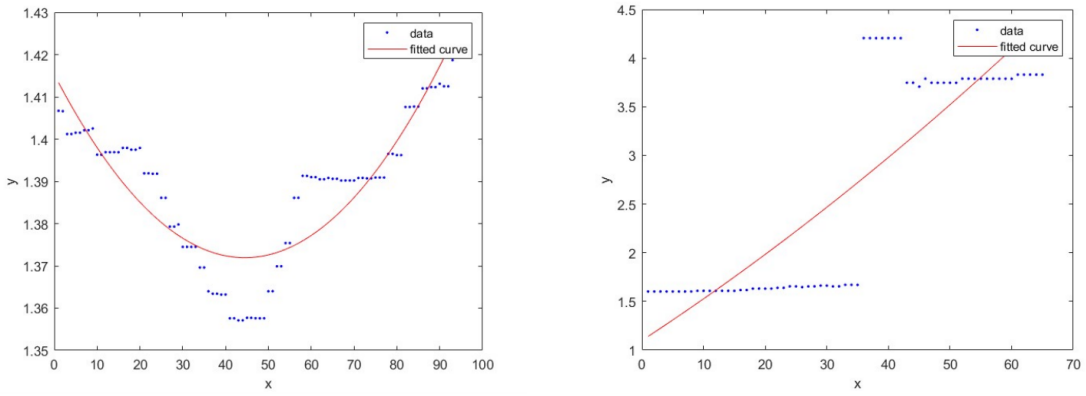
   Il coefficiente marginRate arbitrario indica quanto ci si distanzia dal centro della matrice nel considerare i margini.

   Nel caso in cui il vertice della parabola ottenuta con regressione parabolica si trovi all'esterno del range definito dai due margini allora contrassegno l'immagine come *NonFace*, altrimenti come *Face*.

   La Precision in questo caso varia al variare di *marginRate*

Combinando tutti e 3 i metodi si ha come upperbound della Precision quello del CONCAVITY CHECK, ovvero di circa il 94,6%.

Si può notare che un'immagine Face ha tendenzialmente valori più bassi (quindi più vicini alla camera) verso il centro della matrice (causati dalla presenta della faccia, del naso, ecc.), cosa che non avviene nel caso di una NonFace:

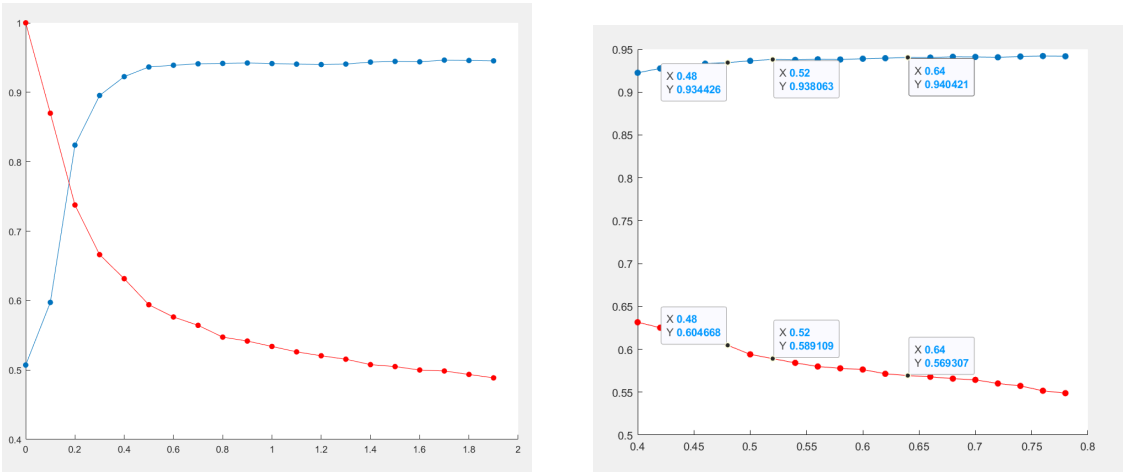


(a) Face (b) NonFace

Figura 1: Confronto di Face e NonFace

Si nota inoltre che la parabola di una Face è giocoforza convessa.

Il marginRate è stato invece scelto sulla base del miglior compromesso tra Precision e Recall.



(a) Precision e Recall con marginRate tra 0 e 2 (b) Precision e Recall con marginRate tra 0.4 e 0.8

Come si può vedere nel grafico all'aumentare del marginRate la Precision aumenta a discapito del Recall.

E' stato scelto per questo come compromesso un marginRate di 0,48 che permette un Recall del 60,4% ed una Precision di 93,4%.

Pseudocodice

```
marginRate = 0.48;
Per ogni elemento con indice i di DepthDATA:
  matrix = DepthDATA{i}{2};
  matrixVCenter = round(size(matrix, 1)/2);
  centralrow = fixedMatrix(matrixVCenter,:);
  clearZeros(centralrow); //rimozione degli zeri
  f = parabolic_fit(centralrow);
  coefficientValues = round(coeffvalues(f), 15);
  vertice = -coefficientValues(2)/(2 * coefficientValues(1)); // -b/2a
  matrixHCenter = round(size(matrix, 2)/2);
  marginA = matrixHCenter - marginRate*size(matrix, 2);
  marginB = matrixHCenter + marginRate*size(matrix, 2);

  if(vertice == NaN or vertice < marginA or vertice > marginB or coefficientValues(1) <= 0):
      //contrassegno come NonFace
  else:
      //contrassegno come Face
```

Analisi Complessità Temporale

Per analizzare la complessità del metodo si tiene conto della complessità temporale delle seguenti funzioni utilizzate:

- size = $O(1)$
- transpose = $O(1)$
- fit = $O(1)$

Per ognuna delle precedenti funzioni sono stati calcolati i vari tempi di esecuzione al variare della quantità di dati utilizzati, i risultati hanno portato alle precedenti conclusioni.

D'ora in avanti considereremo:

- n = numero di immagini
- m = dimensione totale dell'immagine

Il metodo quindi ha complessità temporale pari a $O(n \cdot m^2)$ in quanto per ogni immagine viene calcolato il massimo una volta e poi viene effettuato il controllo degli zeri sulla matrice, valore per valore.

Risultati e Prestazione del metodo

Come detto in precedenza, il metodo ha una Precision e Recall che varia al variare del marginRate. Con il marginRate scelto, ovvero 0,48 si ottiene:

$TruePositive = 855$ (NonFace individuate)

$FalsePositive = 60$ (Face scambiate per NonFace)

$TrueNegative = 1314$ (Face individuate)

$FalseNegative = 559$ (NonFace scambiate per Face)

Il metodo ottiene quindi i seguenti risultati in termini di recall e precision:

$Precision = \frac{TruePositive}{TruePositive+FalsePositive} = \frac{855}{855+60} = 0.9344 \approx 93\%$

$Recall = \frac{TruePositive}{TruePositive+FalseNegative} = \frac{855}{855+559} = 0.6046 \approx 60\%$

Nel caso si vogliano ridurre al minimo i Falsi Positivi (meno di 10), è possibile farlo combinando solo i metodi 1 (PARABOLIC EXISTENCE) e 2 (VERTEX POSITION). Facendo infatti così e usando come marginRate 1,69 si ottiene:

$TruePositive = 375$ (NonFace individuate)

$FalsePositive = 9$ (Face scambiate per NonFace)

$TrueNegative = 1365$ (Face individuate)

$FalseNegative = 1039$ (NonFace scambiate per Face)

Ottenendo:

$Precision = \frac{TruePositive}{TruePositive+FalsePositive} = \frac{375}{375+9} = 0.9765 \approx 98\%$

$Recall = \frac{TruePositive}{TruePositive+FalseNegative} = \frac{375}{375+1039} = 0.2652 \approx 27\%$

Purtroppo in questo caso a discapito della Precision abbiamo un Recall basso.

Legenda file

- metodo1.m contiene l'effettivo programma che controllando tutte le immagini presenti dentro DepthDATA salva nel vettore results se è faccia (0.5) o non faccia (1)
- checkResults.m a partire dal vettore results, conteggia il numero di NonFace correttamente individuate o erratamente individuate
- fixMatrix.m contiene la funziona fixMatrix che data una matrice di profondità sostituisce i valori 0 con il massimo della profondità
- testValori.m testa i vari valori di marginRate (da 0.3 a 0.7) calcolandone di volta in volta Precision e Recall. Alla fine mostra l'andamento del tutto
- risultatiTestValori.fig mostra il grafico risultante dall'esecuzione del file testValori.m con l'andamento di Precision e Recall.