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.

Sui coefficienti del curve fit vengono applicate 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*.

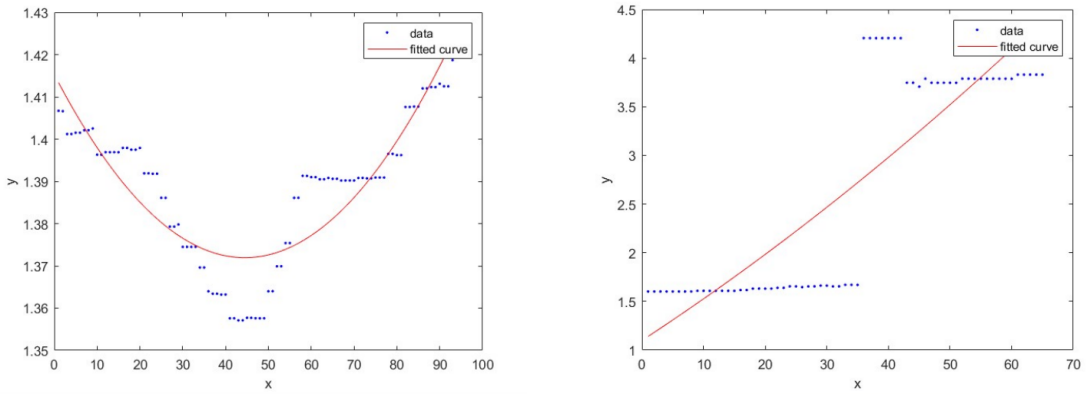
   La Precision in questo caso varia al variare di *marginRate*

I 3 metodi vengono combinati attraverso l'operatore logico *OR*.
Nel caso almeno uno dei 3 metodi identifichi l'elemento come NonFace contrassegno il pattern come *NonFace* altrimenti contrassegno come *Face*.

Combinando i metodi in questo modo, si ha come upperbound della Precision quello del CONCAVITY CHECK, ovvero circa il 94,6%.

Spiegazione metodi

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

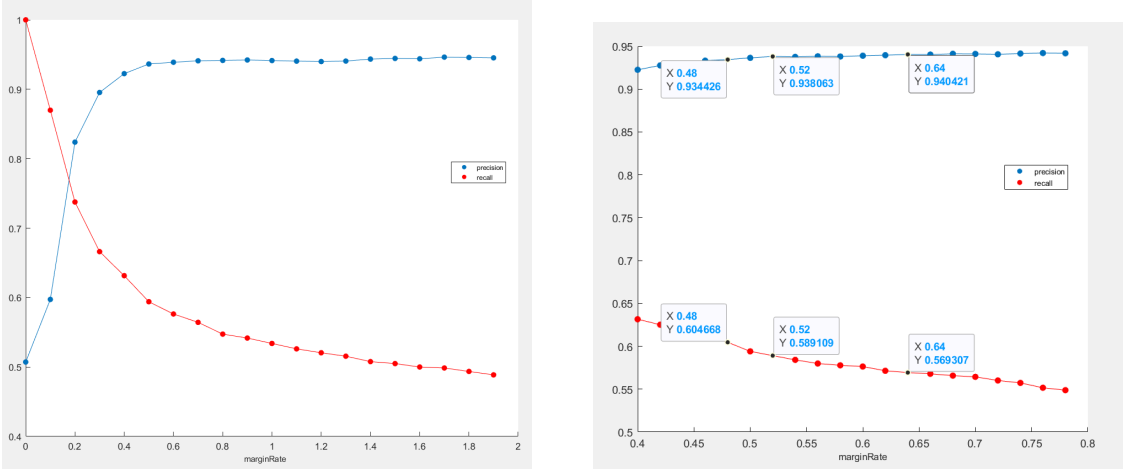
Figura 1: Confronto di Face e NonFace

Questo ci porta ad usare il metodo del curve fit per cercare una parabola che sulla riga centrale della matrice. Il controllo sull'esistenza della parabola porta al controllo **PARABOLIC EXISTENCE**

Si nota inoltre che la parabola di una Face è per forza di cose convessa. Quindi nel caso questa non sia convessa possiamo assumere che sia una NonFace. Da qui il **CONCAVITY CHECK**.

Infine il fatto che la parabola abbia il vertice verso il centro il centro della riga, ci porta al controllo **VERTEX POSITION**.

Il *marginRate* del controllo VERTEX POSITION è stato 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

Figura 2: Scelta marginRate

Come si può vedere nel grafico all'aumentare del *marginRate* la Precision dei 3 metodi usati insieme 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 = matrix(matrixVCenter,:);
  x = 1:size(centralrow,2); //creazione indici
  y = transpose(centralrow);
  clearZeros(x, y); //rimozione degli zeri da y e relativo indice x
  f = parabolic_fit(x, y);
  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 == 0 or vertice ∉ [marginA, marginB] or coefficientValues(1) <= 0):
      //contrassegno come NonFace
  else:
      //contrassegno come Face
```

Analisi Complessità Temporale

Consideriamo:

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

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

- size = $O(1)$
- transpose = $O(1)$
- fit = $O(1)$
- clearZeros = $O(\sqrt[2]{m})$

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.

Il metodo quindi ha complessità temporale pari a $O(n \cdot \sqrt[2]{m})$ in quanto viene effettuato il fit solo sulla riga centrale e non vengono analizzati tutti gli elementi della matrice.

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:

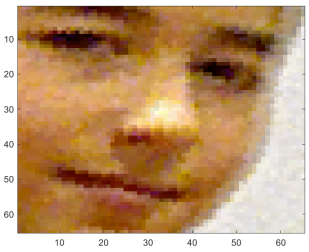
$Precision = \frac{TruePositive}{TruePositive+FalsePositive} = \frac{375}{375+9} = 0.9765 \approx 98\%$

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

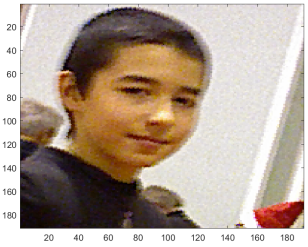
Purtroppo in questo caso la Precision alta è a discapito del Recall.

Analisi degli errori

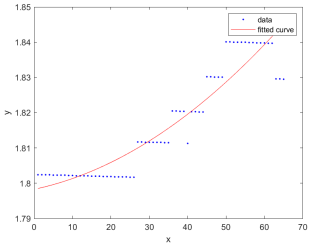
Di seguito vengono mostrate due immagini contrassegnate come False Positive (Face individuate come NonFace).



(a) immagine a colori senza cornice aggiuntiva



(b) immagine a colori dell'area relativa alla depth map con "cornice aggiuntiva"



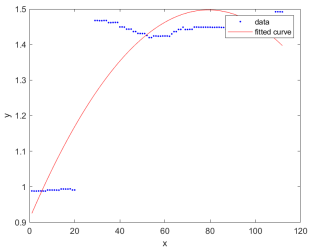
(c) regressione parabolica sulla riga centrale della depthmap senza cornice una volta tolti i valori a 0

Figura 3: Immagine 229

Nella figura 3 si può notare che il metodo non funziona correttamente quando non è presente tutto il viso nell'immagine senza cornice.
In questo caso infatti il vertice della regressione parabolica si trova a circa -15 e passa il **vertex position check** che lo identifica come *NonFace*.

(a) immagine a colori senza cornice aggiuntiva

(b) immagine a colori dell'area relativa alla depth map con "cornice aggiuntiva"

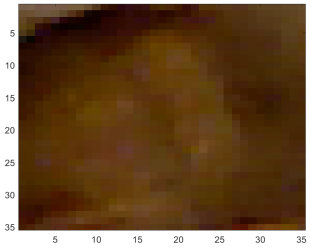


(c) regressione parabolica sulla riga centrale della depthmap senza cornice una volta tolti i valori a 0

Figura 4: Immagine 458

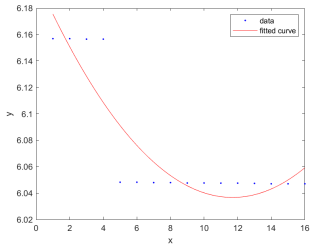
Nella figura 4 si può notare che il metodo non funziona correttamente quando è presente un oggetto che copre parzialmente il volto. In questo caso infatti la regressione parabolica soddisfa il **concavity check** venendo contrassegnata appunto come NonFace.

Ora invece verranno analizzate due immagini contrassegnate come False Negative (NonFace individuate come Face).



(a) immagine a colori senza cornice aggiuntiva

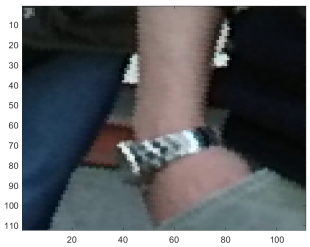
(b) immagine a colori dell'area relativa alla depth map con "cornice aggiuntiva"



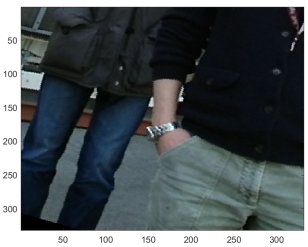
(c) regressione parabolica sulla riga centrale della depthmap senza cornice una volta tolti i valori a 0

Figura 5: Immagine 75

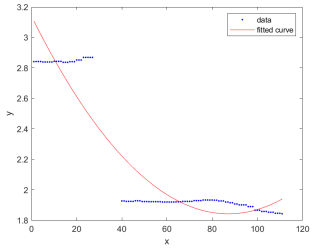
La figura 5 mostra un caso in cui un oggetto di dimensione medio/piccola è posizionato al centro dell'immagine, questo infatti fa in modo che il metodo lo riconosca come Face perché la sua regressione parabolica è pressoché indistinguibile da quella di una vera Face.



(a) immagine a colori senza cornice aggiuntiva



(b) immagine a colori dell'area relativa alla depth map con "cornice aggiuntiva"



(c) regressione parabolica sulla riga centrale della depthmap senza cornice una volta tolti i valori a 0

Figura 6: Immagine 1345

La figura 6 mostra un caso in cui l'immagine a colori senza cornice aggiuntiva prende un'area con un oggetto chiaramente in primo piano, questo fa in modo che venga individuata come Face vista la sua regressione parabolica.

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
- 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.