



# Disparity e V-disparity

Corso di Fondamenti di  
Visione Artificiale  
AA 2020/2021

# Input

- Immagini stereo
  - R.pgm e L.pgm
  - Già rettificate




# Esercizio #1

- Calcolare l'immagine di disparità usando SAD - Sum of Absolute Distances
  - Usare finestra  $W \times W$  con  $W$  passato come argomento
  - Dato pixel  $(r, c)$  su  $L.pgm$ 
    - Confrontare vicinato  $W \times W$  con vicinati possibili candidati destra
    - Dove la somma delle differenze in valore assoluto è minore  $\rightarrow$  candidato
    - Salvare valore di disparità  $d$  in immagine di uscita in posizione  $(r, c)$
    - Limitare range di ricerca di  $d$  tra  $[0, 127]$



# Esercizio #1

- Usare la seguente funzione:




- void SAD\_Disparity(  
    const cv::Mat &l,  
    const cv::Mat &r,  
    unsigned short w\_size,  
    cv::Mat & out); 

- Esempio di esecuzione

- ./simple ../images/L.pgm ../images/R.pgm 7

# Esercizio #1

## ■ Hints

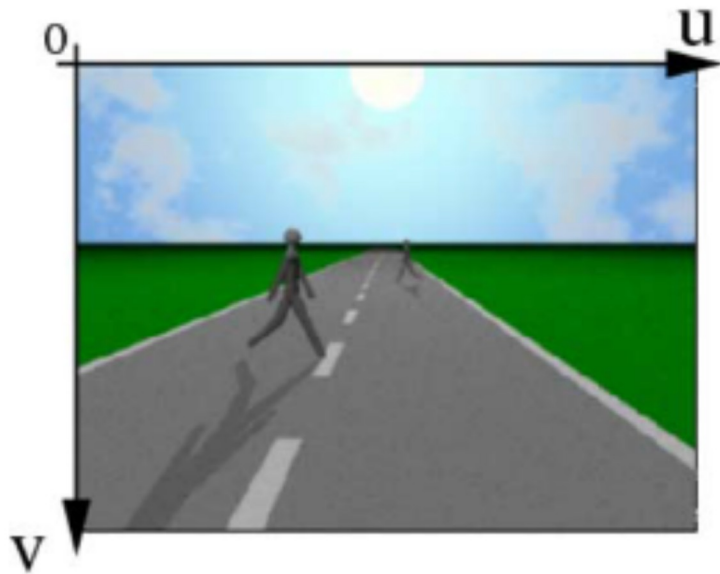
- su quale riga dell'immagine destra devo cercare il corrispondente del pixel di destra?   
Le immagini sono rettificate, quindi i piani immagine sono paralleli e complanari.
- una volta nota la riga, quali colonne ha senso esplorare? Quelle di destra o quelle di sinistra? 
- si puo' fare con 5 cicli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



# V-Disparity

- La V-Disparity è una particolare tecnica “semplificata” di confronto stereoscopico
- Le righe delle due immagini vengono confrontate contando per ogni possibile disparità il numero di pixel che corrispondono e creando un istogramma
- Quindi un istogramma per riga → nuova immagine con lo stesso numero di righe e numero di colonne uguale a intervallo disparità considerate

# V-Disparity



Left image






Right image



V-disparity



# V-Disparity

- Due approcci
  - Parto da immagine disparità → maggior precisione 
  - Direttamente confronto pixel a pixel → più veloce 
- Il risultato può essere ulteriormente elaborato estruendo
  - Retta obliqua → informazione su orientazione piano
    - Image stabilization 
  - Segmenti verticali → informazioni su ostacoli



# Esercizio #2



- Usando il risultato dell'esercizio precedente:
  - Creare una nuova immagine `vdisp` di larghezza pari a 128 in cui si codifichi riga per riga la V-disparity tra le due immagini in ingresso
  - Usare come prototipo:
    - `void VDisparity(const cv::Mat & disp, cv::Mat & out)`

# Esercizio #3



- Calcolare l'immagine di V-Disparity direttamente dalle due immagini in ingresso
- Usare un confronto pixel a pixel
  - `void V_Disparity(  
    const cv::Mat &l,  
    const cv::Mat &r,  
    cv::Mat & out);`