UNIVERSITÀ DEGLI STUDI
DI MILANO

Informazione Multimediale - Video
Elaborazione delle Immagini

Raffaella Lanzarotti

OPTICAL FLOW

First step: open a video file

```
OBJ = VideoReader(FILENAME);
```

- Construct an obietto read video
- Associated methods:
  - `readFrame` - Read next frame
  - `hasFrame` - Determine if there are other available frames to be read

Object properties

- `Name` - file name.
- `Path` - file path
- `Duration` - file length in seconds
- `CurrentTime` - Current frame position (expressed in seconds)
- `Height` - frame height (in pixel)
- `Width` - frame width (in pixel)
- `BitsPerPixel`
- `VideoFormat`
- `FrameRate` - (in frame per secondo)

Optical flow (LK)

```
obj = opticalFlowLK
```

- Return an object to evaluate the optical flow by the Lucas Kanade method.
- The optical flow is expressed in terms of direction and velocity from previous to current frame
- methods:
  - `estimateFlow` - estimate of optical flow
  - `reset` - Reset of the object internal state
- Property:
  - `'NoiseThreshold'`: threshold for noise reduction. the higher it is the less the computation is sensible to small movements. Default: 0.0039

Point Tracking, KLT

- Given some points on the first frame, we want to track them.
- Problem: which points?
  - we could choose good points according to heuristics (Harris, SIFT, …) but they would not guarantee good tracking

- IDEA of Kanade-Tomasi:

A feature is good if it can be well tracked.

Recall Lucas Kanade

$$\begin{bmatrix} \sum f_{xi}^2 & \sum f_{xi} f_{yi} \\ \sum f_{xi} f_{yi} & \sum f_{yi}^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -\sum f_{xi} f_{ti} \\ -\sum f_{xi} f_{ti} \end{bmatrix}$$

$A^T A$

- Which is the condition of good conditioning?
  - $A^T A$ should be <u>invertible</u>
  - $A^T A$ not too small
    - eigenvalues (λ_1, λ_2) not too small
  - $A^T A$ well conditioned

    λ_1 / λ_2 not too big, where $\lambda_1 > \lambda_2$

Point Tracking, KLT, cont.

- a point is good if

$$\min(\lambda_1, \lambda_2) > \lambda$$

- Which threshold value?

- theoretically, disposing of the acquisition camera, we could measure the eigenvalues corresponding to a uniform region (characterize the camera noise) —> lower bound λ_{low}
- measure the eigenvalues of regions with high texture variations —> upper bound λ_{high}
- fix the threshold in the middle

$$\lambda = 0.5 * \lambda_{low} + 0.5 * \lambda_{high}$$

Point Tracking, MATLAB

```
H = vision.PointTracker
```

- Create an object, H to track a set of points using the Kanade-Tomasi feature tracking algorithm

- It requires an initialization to specify the initial positions of points on the initial frame:

```
initialize(H, POINTS, I)
```

Point Tracking, cont.

- Frame by frame tracking is obtained using the method **step**

```
[POINTS,POINT_VALID,SCORES] = step(H, I)
```

- `POINTS`: array MX2 of coord [x y] corresponding to new coordinates
- `POINT_VALID`: array Mx1, of boolean values, saying if the found point is reliable or not
- `SCORES`: array Mx1, of real values between 0 and 1, computed as LSE among the blocks around the tracked point

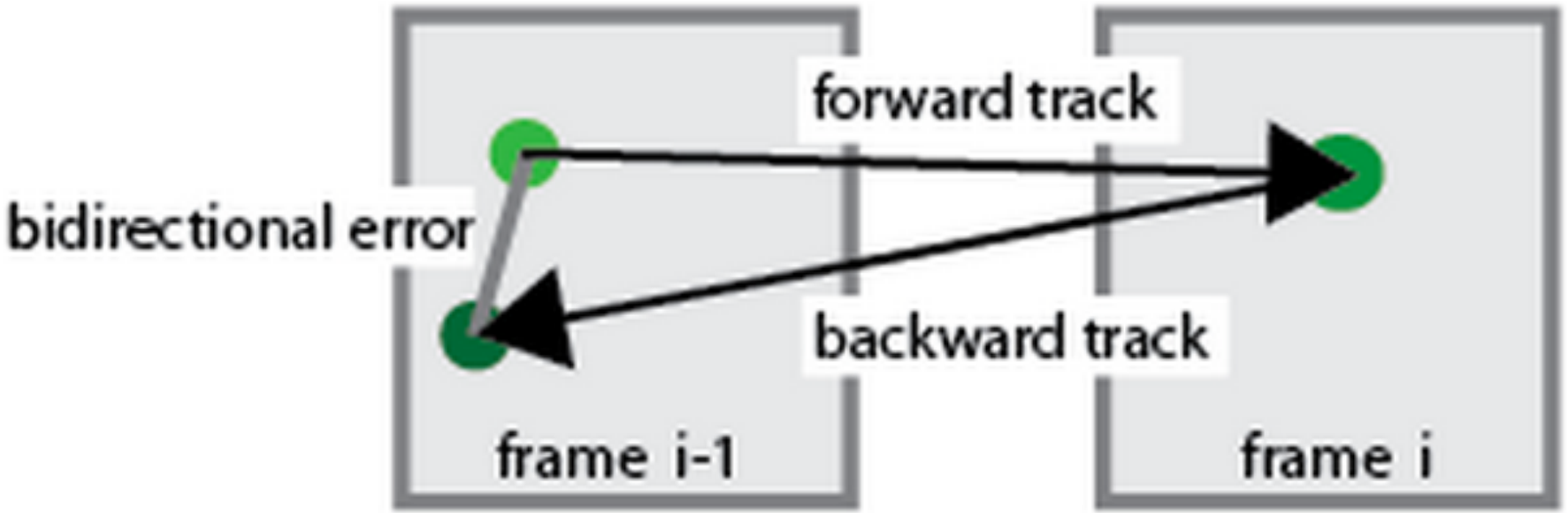
- It is possible to reset the points:

```
setPoints(H, POINTS)
```

Point Tracker Properties

- `BlockSize` - dimension of the integration window to be centered around each point
- `NumPyramidLevels` - number of pyramid level
- `MaxBidirectionalError` - Max threshold of the forward-backward error.

- `MaxBidirectionalError`,
  ideal in the range 0-3



Default: `inf`