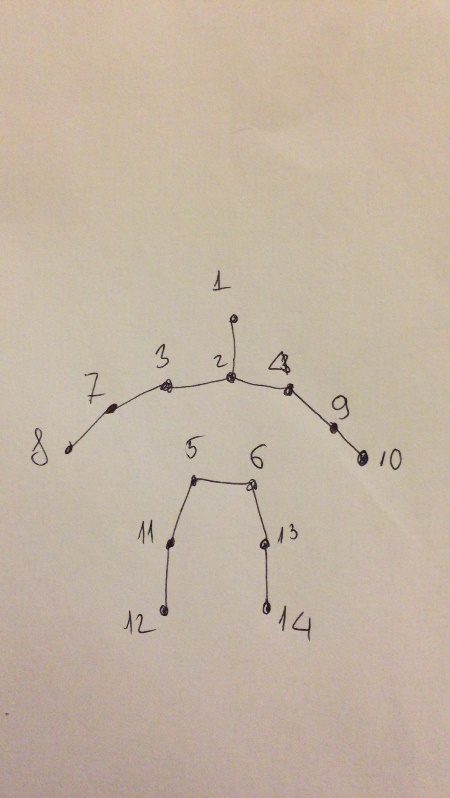
1. **Introduction**

The toolkit is written in Python (ver 2.7) with Tkinter for UI and Opencv for Video/Image processing. It contains two functions:

* Manually edit joints in a completed 14-joints-skeleton;
* Automatically track given joints, extract key frames.

1. **Functions**
   1. *Manually edit joints*

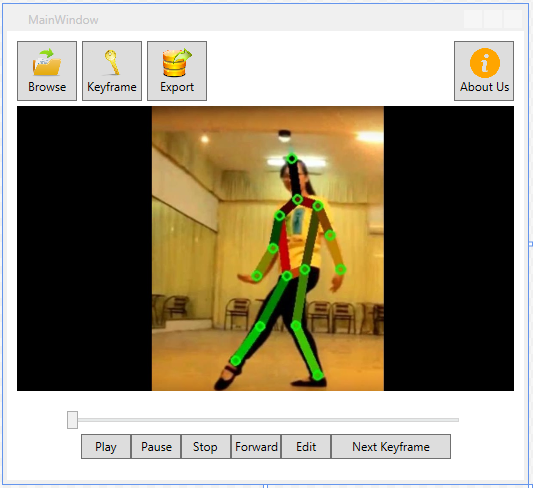
The toolkit supports different video format/human-pose format; manually edits joint points; exports to different format files.

* Input: a 2D video clip, and corresponding skeleton file. The people exist in the video, correspoding skeleton file saves the coordinates of 14-joints-skeleton. Human skeleton is created by correspondingly connecting the 14 joints (Fingure 1).

**Fingure 1**: Human skeleton with 14 joints

* Output: a single skeleton file.

Example about UI of the toolkit (Fingure 2): The found joints are marked with green circles. These joints can be moved to right position and save in skeleton file.

 **Fingure 2**: The toolkit’s UI

* 1. *Automatically track given joints, extract key frames*

This is an advance feature of the toolkit: extract key frames which have significant changes in motions or have high possibility of pose estimation errors, we focus on the first type of key frame. As a result, the time for editing video is down to 2-3 times.

* Input: a 2D video clip, manually pick 14 point on the first frame
* Output: a 2D video clip contains skeleton movement of the performer.
* There are two methods for key frame extraction and tracking joints:
  + Using Color-histogram difference:

**function** keyframeExtraction

**input**: S frames of video, T frames desired

**output**: indices of keyframes

**begin**

create array X

for i = 0 to N-1 do

K <- convert frames[i] to gray

Kplus <- convert frames[i+1] to gray

m <- calcHist(K, 256)

n <- calcHist(Kplus, 256)

sss <- sum(abs(m-n))

append sss to array X

end for

I <- argsort(X) + 1

J <- I[len(X)- T : len(X)]

Return sort(J)

**End**

* + Optical flow:

**function** trackPoint()

**Input**: point p0 in first frames, N frames of video,

**Output**: position of points in video

**Begin**

Create array trackedframes, POSITION

Mark all joints with circle at p0

Append p0 to POSITION

Append the first frame to trackedframes

For i = 0 to N-1 do

frame <- ith frame

p1, st, err <- calcOpticalFlowPyrLK(old\_gray, frame\_gray, p0, None, \*\*lk\_params)

if all stj­ == 1 and err­j < threshold do

mark all joint with circle at p1

append frame to trackedframes

append p1 to POSITION

end for

return POSITION

**End**