Center of Mass Module Manual - TEEP

Elvis Rodrigues, UCLA

1 Introduction

This document provides a brief explanation on the center of mass module and its math. The COM module was written for the PyTorch implementation of lightweight-openpose by Daniil Osokin found here. The module exploits the point output from the 'fast-inference' function in `lightweight-openpose/demo.py` and computes the center of mass and base of support for each pose in an image or video frame.

This is done in 3 stages: heuristics, transform and compute. The math in this module is based on the following anthropometry table.

Table 3.1 Anthropometric Data

Segment	Definition	Segment Weight/ Total Body Weight	Center of Mass/ Segment Length		Radius of Gyration/ Segment Length			Density
			Proximal	Distal	C of G	Proximal	Distal	
Hand	Wrist axis/knuckle II middle finger	0.006 M	0.506	0.494 P	0.297	0.587	0.577 M	1.16
Forearm	Elbow axis/ulnar styloid	0.016 M	0.430	0.570 P	0.303	0.526	0.647 M	1.13
Upper arm	Glenohumeral axis/elbow axis	0.028 M	0.436	0.564 P	0.322	0.542	0.645 M	1.07
Forearm and hand	Elbow axis/ulnar styloid	0.022 M	0.682	0.318 P	0.468	0.827	0.565 P	1.14
Total arm	Glenohumeral joint/ulnar styloid	0.050 M	0.530	0.470 P	0.368	0.645	0.596 P	1.11
Foot	Lateral malleolus/head metatarsal II	0.0145 M	0.50	0.50 P	0.475	0.690	0.690 P	1.10
Leg	Femoral condyles/medial malleolus	0.0465 M	0.433	0.567 P	0.302	0.528	0.643 M	1.09
Thigh	Greater trochanter/femoral condyles	0.100 M	0.433	0.567 P	0.323	0.540	0.653 M	1.05
Foot and leg	Femoral condyles/medial malleolus	0.061 M	0.606	0.394 P	0.416	0.735	0.572 P	1.09
Total leg	Greater trochanter/medial malleolus	0.161 M	0.447	0.553 P	0.326	0.560	0.650 P	1.06
Head and neck	C7-T1 and 1st rib/ear canal	0.081 M	1.000	— PC	0.495	1.116	— PC	1.11
Shoulder mass	Sternoclavicular joint/ glenohumeral axis	—	0.712	0.288	—	—	—	1.04
Thorax	C7-T1/T12-L1 and diaphragm*	0.216 PC	0.82	0.18	—	—	—	0.92
Abdomen	T12-L1/L4-L5*	0.139 LC	0.44	0.56	—	—	—	—
Pelvis	L4-L5/greater trochanter*	0.142 LC	0.105	0.895	—	—	—	—
Thorax and abdomen	C7-T1/L4-L5*	0.355 LC	0.63	0.37	—	—	—	—
Abdomen and pelvis	T12-L1/greater trochanter*	0.281 PC	0.27	0.73	—	—	—	1.01
Trunk	Greater trochanter/ glenohumeral joint*	0.497 M	0.50	0.50	—	—	—	1.03
Trunk head neck	Greater trochanter/ glenohumeral joint*	0.578 MC	0.66	0.34 P	0.503	0.830	0.607 M	—
HAT	Greater trochanter/ glenohumeral joint*	0.678 MC	0.626	0.374 PC	0.496	0.798	0.621 PC	—
HAT	Greater trochanter/mid rib	0.678	1.142	—	0.903	1.456	—	—

*NOTE: These segments are presented relative to the length between the greater trochanter and the glenohumeral joint.

Source Codes: M, Dempster via Miller and Nelson; *Biomechanics of Sport*, Lea and Febiger, Philadelphia, 1973. P, Dempster via Plagenhoef; *Patterns of Human Motion*, Prentice- Hall, Inc. Englewood Cliffs, N.J., 1971. L, Dempster via Plagenhoef from living subjects; *Patterns of Human Motion*, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1971. C, Calculated.

The lines to focus on are 'Forearm', 'Upper arm', 'Thigh', 'Foot and leg', 'Total leg' and 'Trunk head neck'. The relevant columns are 'Segment Weight / Total Body Weight' (the ratio of the segment's mass to total mass) and 'Segment Length' (which gives the position of a segment's center of mass).

To explain the segment length column, one needs to know what is the Proximal point and Distal point of a segment. 'Proximal' roughly refers to the point closest to the 'center' of the body and 'Distal' is the opposite. Consider the upper arm: the table defines this segment with the Proximal shoulder (glenohumeral joint) and Distal elbow. The upper arm's center of mass is located at 0.436 times the length from the elbow to the shoulder:

2 Math

The math involved is fairly simple. Since an image is 2D, we must compute the center of mass as an (x, y) coordinate pair. Suppose we have computed the Segment COM coordinates from the anthropometry table above. Then the x-coordinate of the final center of mass is:

$$\mathrm{COM}_x = \frac{1}{M}\sum_{i=1}^{N} \mathrm{com}_{i,x} \cdot m_i = \sum_{i=1}^{N} \mathrm{com}_{i,x} \cdot \left(\frac{m_i}{M}\right)$$

where M is the person's total mass, N is the number of body segments used (5 in our case), $\mathrm{com}_{i,x}$ is the x-coordinate of the i-th segment center of mass and m_i is the mass of the i-th segment. Notice that m_i/M is the Segment Weight Ratio from the table above for the i-th segment.

The base of support is estimated from the ankle point and defines the area of stability where a vertically-projected Center of Mass must stay in for stability. This method of estimation is subject to change soon.

3 Lightweight Openpose

The PyTorch implementation of lightweight-openpose finds and appropriately labels 17 points of the body in a 'Pose' object.

The image above is an example of a single pose with all 17 points. The center of mass module connects these points and uses their vectors to locate the segment's center of mass. The minimum requirements asserted to find the final center of mass is

1. either the set {Right Shoulder, Right Elbow, Right Wrist, Right Hip, Neck}
2. or the set {Left Shoulder, Left Elbow, Left Wrist, Left Hip, Neck}
3. Additionally, the point set must include at least one of [Nose, Right Ear, Left Ear].

4 Heuristics - lines 49 to 68

This part of the code checks if its requirements are satisfied. It checks if the points from the sets mentioned above are available. If both condition 1 and 2 are met, then we are processing the Coronal Plane (front/back) of the human body. If either condition 1 or 2 are met but not both, then we are processing the Sagittal Plane (sides) of the human body. Heuristics also picks the first point that satisfies condition 3: if the Nose point is available, it is used regardless of the Ear points.

Note that there are no leg points like the knees and ankles in the conditions above. This is to allow the module to estimate COM if the legs are missing from the image. If the knees and ankles are missing, the module uses the Nose/Ear-Neck vector to estimate the leg COM: a human leg is 3.5 to 4 heads long. In the code comments, the Proximal ratios used when compensating for missing leg points are labeled 'magic numbers' since they factor in the Nose/Ear-Neck vector.

5 Transformation - lines 70 to 86

This part of the module handles missing points if the human pose is from the Sagittal Plane. The assumption made here is that if person is facing sideways and a point is missing, then its corresponding pair can be copied over to it. For example, if we have an image of a person's right side and the Left Knee point is missing, then we copy the coordinates of the Right Knee point to the Left Knee point.

If the image is from the Coronal Plane, then nothing is done.

6 Compute Segment COM - lines 88 to ??

This part is subject to change soon for better Base of Support estimation. Most of the computation is done here. The point pairs are connected to each other here - this should be changed for easier configuration. The segment COM is then computed by adding the connection vector scaled by the Proximal ratios. These points are collected in the list `C_pts`:

```
mass_ratios = array([[0.578, 0.028, 0.028, 0.022, 0.022, 0.1, 0.061, 0.161, 0.1, 0.061, 0.161]])
mass_ratios = tile(mass_ratios.transpose(), (1, 2))
proximal_ratios = [0.66, 0.436, 0.43, 0.433, 0.606, 0.69, 2.07] # final 2 are magic factors for missing points
```

Figure: Proximal Length Ratios and Mass Ratios

The method of estimating Base of Support is subject to change - please refer to code comments. In case of missing leg points, the module takes the Nose/Ear-Neck vector and greedily estimates the leg (lower half) or total leg COM by scaling the vector with 'magic numbers'. The magic numbers were computed by multiplying the Head vector with the proximal ratio for a total leg or just the lower leg.

Note that in some places, I switched Proximal for Distal ratios for better accuracy: mostly for the legs. I will try to label these.

7 Compute Final COM and BOS - lines ?? to ??

This part is subject to change soon for better Base of Support estimation. The segment COM points are collected and multiplied with the mass ratios. Once they are all added together, we have the final COM. This part implements the math discussed in Section 2.

Base of Support uses some 'soft' estimation of the foot when needed. This part needs some reworking to ensure guarantees of the COM module are met. The module finally returns a tuple of (1) the Final COM, (2) the Segment COMs and (3) the Final BOS points pair with the first guaranteed to be the 'left' line and the second guaranteed to be the 'right' line.