CSCI473/573 Human-Centered Robotics
Project 2: Skeleton-Based Representations for Robot Understanding of Human Behaviors

Project Assigned: February 21
Multiple due dates (all materials must be submitted to Canvas)
Deliverable 1 (Representation Code) due: March 8, 23:59:59
Deliverable 2 (Complete Code) due: March 15, 23:59:59
Deliverable 3 (Project Report) due: March, 18, 23:59:59
*****Note: This version only contains the requirements of Deliverable 1*****

In this project, students will implement several skeleton-based representations (Deliverable 1) and use Support Vector Machines (SVMs) (Deliverable 2) to classify human behaviors using a public activity dataset collected from a Kinect V1 sensor. Additionally, students are required to write a report following the format of standard IEEE robotics conferences using LaTeX as Deliverable 3.

Students are required to program this project using C++ or Python in a Linux system, although students are not required to implement the project in ROS. Your code must be able to run on Ubuntu 16.04 LTS, Ubuntu 14.04 LTS, or the Linux machines to program the project.

I. Dataset

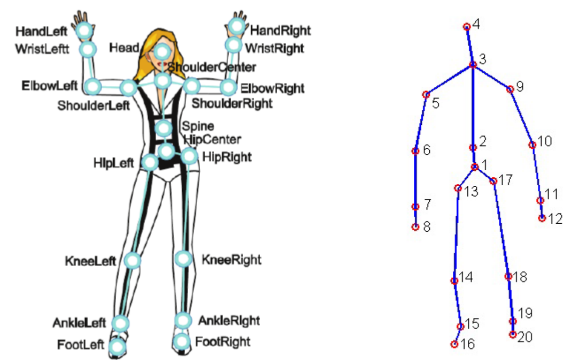
The MSR Daily Activity 3D dataset[1] will be used in this projet, which was one of the most widely applied benchmark dataset in human behavior understanding tasks. This dataset contains 16 human activities, as demonstrated in Figure 2, performed by 10 human subjects. Each subject performs each activity twice, once in a standing position, and the other in a sitting position. Although this dataset contains both color-depth and skeleton data, in this project, we will only explore the skeletal information to construct **skeleton-based human representations** (Deliverable 1).

The skeleton in each frame contains 20 joints, as illustrated by Figure 1. The correspondence between joint names and joint indices is also presented in the figure. For example, Joint #1 is HipCenter, Joint #18 is KeeRight, etc.

In this project, a pre-formatted skeleton data will be used, which can be downloaded in the course website.
`http://inside.mines.edu/~hzhang/Courses/CSCI473-573/assignment.html`.

This dataset in this project is a subset of original dataset, which only contains six (6) activity categories:

- CheerUp (`a08`)



(a) Names (b) Indices

Fig. 1: Skeleton joint names and indices from Kinect SDK.

- TossPaper (`a10`)
- LieOnSofa (`a12`)
- Walk (`a13`)
- StandUp (`a15`)
- SitDown (`a16`)

It is also noteworthy that this dataset is not formatted to the LIBSVM format but much easier to process then the original data format. Conversion of the data to the SVM format will be a task for Deliverable 2.

In particular, the dataset contains two folders: `Train` and `Test`. Data instances in the directory `Train` is used for training (and validation). Data instances in `Test` is used for testing the performance. Each instance has a filename like: `a12_s08_e02_skeleton_proj.txt`. This filename means the data instance belongs to activity category 12 (i.e., `a12`, that is "lie down on sofa" as in Figure 2), from human subject 8 (`s08`) at his/her second trial (i.e., `e02`). The dataset contains 16 activity categories, 10 subjects, and 2 trials each subject. Instances from subjects 1–6 are used for training (in the directory `Train`), and instances from subjects 7–10 are used for testing (`Test`).

When you open a data instance file, say `a12_s08_e02_skeleton_proj.txt`, you will see something like:

If you have any questions or comments, please contact the instructor Prof. Hao Zhang at `hzhang@mines.edu`

This write-up is prepared using LaTeX.

[1]The MSR Daily Activity 3D dataset is removed from the author's website and no longer publicly available after 2017.

1	1	0.326	-0.101	2.111
1	2	0.325	-0.058	2.152
1	3	0.319	0.194	2.166

.
.
.

So each *row* contains five values, representing:

1) `frame_id`,
2) `joint_id`,
3) `joint_position_x`,
4) `joint_position_y`,
5) `joint_position_z`.

Each frame contains 20 rows that contain information of all joints in the frame.

II. **CSCI 473**: Deliverable 1 (Representation Construction)

Students in CSCI 473 must implement two skeleton-based representations during the Deliverable 1.

A. *Relative Distances and Angles of Star Skeleton*

Students in CSCI 473 are required to implement the human representation based on the **Relative Angles and Distances (RAD) of star skeleton**, as described by Algorithm 1. The objective is to implement the RAD representation to convert all data instances in the folder `Train` into a single training file `rad_d1`, each line corresponding the RAD representation of a data instance. Similarly, all instances in the folder `Test` needs to be converted into a single testing file `rad_d1.t`.

B. *Customized Representations*

Implement at least a new skeleton-based representation by choosing different joints other than the joints selected in the star skeleton. For example, you can change reference joints, use other joints other than body extremities, or compute distances of all joints but ignore the orientation information, as the representation of *Histogram of Joint Position Differences* (HJPD). Your code is required to output a single training file `cust_d1` for all training instances, with each row containing the customized representation of an instance, and a single testing file `cust_d1.t`, similar to the task in Section II-A.

C. *What to Submit*

For the Deliverable 1, CSCI 473 students are required to submit a **single tarball**, named *T1_firstname_lastname.tar* (or .tar.gz) to the Blackboard portal named P2-T1, which must contain the following items:

- A README that provides sufficient instructions needed to compile and execute your code. Your README also needs to document your implementation information, including which joints are used in the RAD representation, how the histograms are computed, and how many bins are used.
- Your code to construct the RAD and customized representations.
- The generated representation data, including `rad_d1`, `rad_d1.t`, `cust_d1`, and `cust_d1.t`.

Algorithm 1: RAD representation using star skeletons

Input : Training set `Train` or testing set `Test`
Output : `rad_d1` or `rad_d1.t`

for *each instance in* `Train` *or* `Test` **do**
    for *frame* t = *1, ..., T* **do**
        Select joints that form a star skeleton (Figure 3);
        Compute and store distances between body extremities to body center ($d_1^t, ..., d_5^t$);
        Compute and store angles between two adjacent body extremities ($\theta_1^t, ..., \theta_5^t$);
    end
    Compute a histogram of N bins for each $\boldsymbol{d}_i = \{d_i^t\}_{t=1}^T$, $i = 1, ..., 5$;
    Compute a histogram of M bins for each $\boldsymbol{\theta}_i = \{\theta_i^t\}_{t=1}^T$, $i = 1, ..., 5$;
    Normalize the histograms by dividing T to compensate for different number of frames in a data instance;
    Concatenate all normalized histograms into a one-dimensional vector of length $5(M+N)$;
    Convert the feature vector as a single line in the `rad_d1` or `rad_d1.t` file.
end
return *`rad_d1` or `rad_d1.t`*

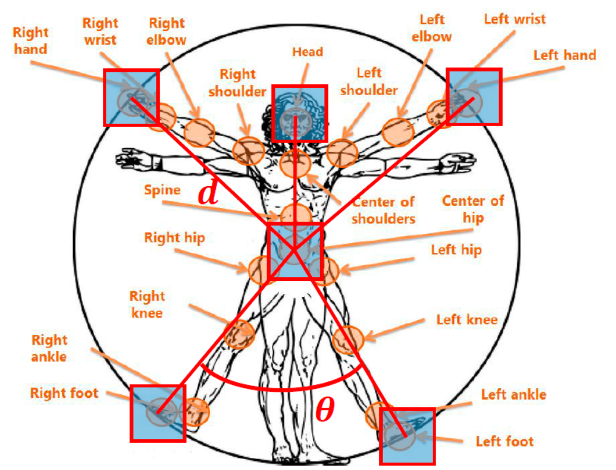


Fig. 3: Illustration of human representation based on relative distance and angles of star skeleton

Students are allowed to include a local copy of the training and testing sets within the code directory to make your code-self contained.

III. **CSCI 573**: Deliverable 1 (Representation Construction)

CSCI 573 students are required to implement three specific skeleton-based representations for the Deliverable 1, including RAD, HJPD, and HOD.

A. *Relative Distances and Angles of Star Skeleton*

Students in CSCI 573 are required to implement the human representation based on the **Relative Angles and Distances (RAD) of star skeleton**, as described by Algorithm 1. The objective is to implement the RAD representation to convert all data instances in the folder `Train` into a single training file `rad_d1`, each line corresponding the RAD representation

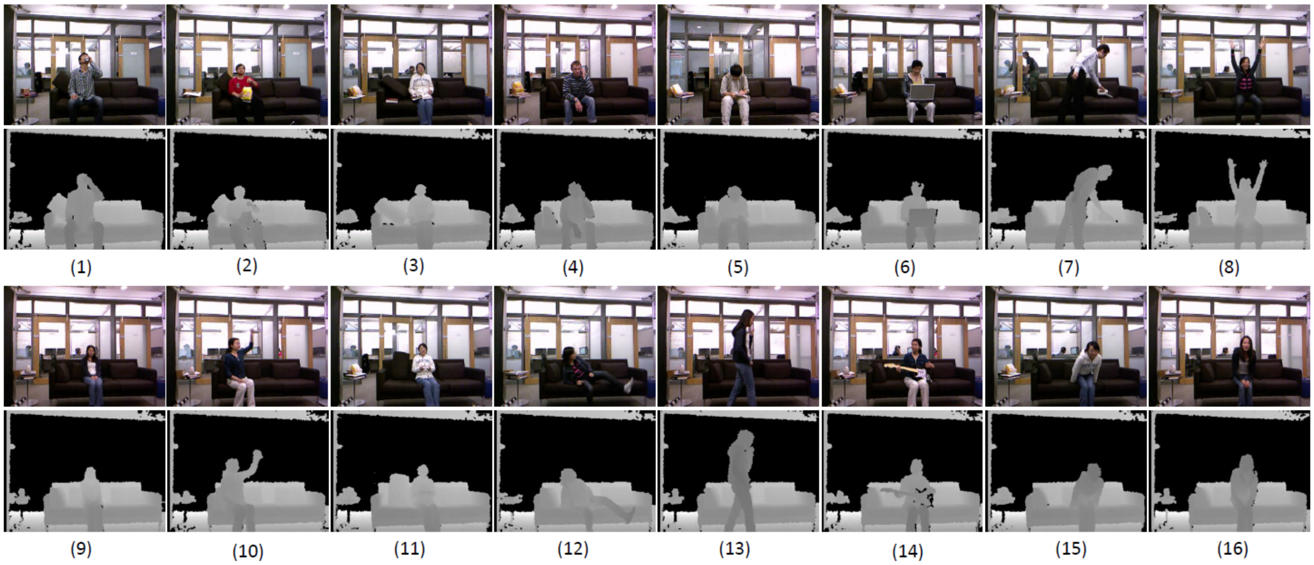


Fig. 2: The full MSR Daily Activity 3D dataset contains sixteen human activities: (1) drink, (2) eat, (3) read book, (4) call cellphone, (5) write on a paper, (6) use laptop, (7) use vacuum cleaner, (8) cheer up, (9) sit still, (10) toss paper, (11) play game, (12) lie down on sofa, (13) walk, (14) play guitar, (15) stand up, (16) sit down.

of a data instance. Similarly, all instances in the folder `Test` needs to be converted into a single testing file `rad_d1.t`. This required representation is the same as Section II-A.

B. Histogram of Joint Position Differences (HJPD)

Given the 3D location of a joint (x, y, z) and a reference joint (x_c, y_c, z_c) in the world coordinate, the joint displacement is defined as:

$$(\Delta x, \Delta y, \Delta z) = (x, y, z) - (x_c, y_c, z_c) \tag{1}$$

The reference joint can be the skeleton centroid or a fixed joint. For each temporal sequence of human skeletons (in a data instance), a histogram is computed for the displacement along each dimension, i.e., $\Delta x, \Delta y, \Delta z$. Then, the computed histograms are concatenated into a single vector as a feature.

This HJPD representation is similar to the RAD representation, except that it uses all joints and ignores the pairwise angles. Refer to Section 3.3 of reference [1] for more details, which is available online at:

`http://staffhome.ecm.uwa.edu.au/~00053650/papers/hossein_WACV2014.pdf`.

Your code is required to generate a single training file `hjpd_d1` for all training instances, with each row containing the HJPD representation of an instance, and a single testing file `hjpd_d1.t`.

C. Histogram of Oriented Displacements (HOD)

You need to implement the skeleton-based representation of Histogram of Oriented Displacements (HOD), as introduced in Section 3 of reference [2], including the Temporal Pyramid. The paper is available online at:

`http://www.aaai.org/ocs/index.php/IJCAI/IJCAI13/paper/view/6967`.

Your code is required to generate a single training file `hod_d1` for all training instances, with each row containing the HOD representation of an instance, and a single testing file `hod_d1.t`.

D. What to Submit

For the Deliverable 1, CSCI 573 students are required to submit a **single tarball**, named *T1_firstname_lastname.tar* (or .tar.gz) to the Blackboard portal named P2-T1, which must contain the following items:

- A README that provides sufficient instructions needed to compile and execute your code. Your README also needs to document your implementation information, for example, including which joints are used in the RAD representation, and how the histograms are computed and how many bins are used in your HJPD and HOD representations.
- All your code to construct the RAD, HJPD, and HOD representations.
- All the generated skeleton-based representation data, including `rad_d1`, `rad_d1.t`, `hjpd_d1`, `hjpd_d1.t`, `hod_d1`, and `hod_d1.t`.

Students are allowed to include a local copy of the training and testing sets within the code directory to make your code-self contained.

IV. Grading

Your grade will be based upon the quality of your project implementation and the documentation of your findings in the report.

- 30%: The quality of the Deliverable 1. You should have a "working" software implementation, which means that your skeleton-based representations are implemented in software, your code runs without crashing, and performs the behavior understanding task.

Students in CSCI573 will be graded more strictly on the quality of the code implementation.

REFERENCES

[1] H. Rahmani, A. Mahmood, D. Q. Huynh, and A. Mian, "Real time action recognition using histograms of depth gradients and random decision forests," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2014.

[2] M. A. Gowayyed, M. Torki, M. E. Hussein, and M. El-Saban, "Histogram of oriented displacements (hod): Describing trajectories of human joints for action recognition.," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2013.