Action Recognition for a Job Coach System in Vocational Rehabilitation

1. Introduction

Individuals with cognitive disabilities face great difficulty in education and vocational rehabilitation. Nevertheless, research results indicate that traditional "brain-train" approaches are ineffective in addressing the impact of cognitive disability in general. A “job coach” system is set up to provide direct interventions and feedback to address complex work problems, in order to provide a natural environment for vocational rehabilitation for an individual with a disability. However, such "one-to-one" job coaching is often costly and inefficient.

This project is proposed to develop an automatic and remote job coach system that uses the task of cooking hamburgers as a test bed (Fig. 1), which can possibly reduce the involvement of a full-time "coach". The goal is to provide guidance to an employee with a cognitive disability who has difficulty remembering and reproducing a long sequence of actions. The system would provide support in a training or a work environment by recognizing the necessary steps in cooking hamburgers and providing guidance when a step is missed. It is done by using motion capture data with limited number of markers, and distinguishing different hamburger cooking actions from these data.

2. Method: Example-based Recognition

The first stage is using example-based recognition. This is done by using motion capture data with 3 reflective markers on each hand (6 markers in total), markers on the grill and the spatula for training and testing (Fig. 2). The data is hand-labeled as different primitive actions (placing, flipping, salting, pressing and picking).

Figure 1: The first job coach system using the task of cooking hamburgers.

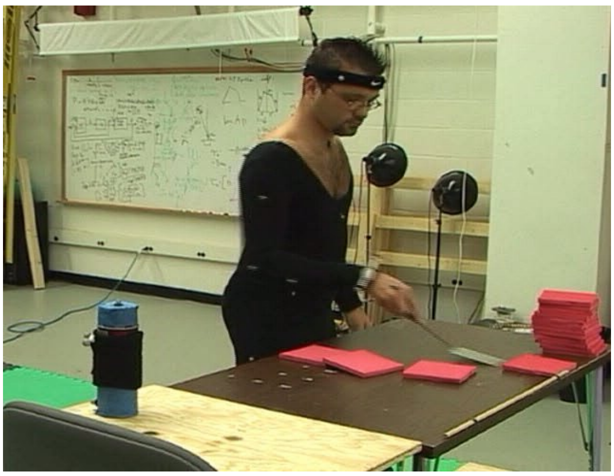

Figure 2: Motion capture of cooking hamburgers using fake hamburgers.

3D positions are given by the motion capture data, therefore different features such as velocities, accelerations and hand orientations can be computed. After comparing features over a window of time, a *k*-nearest neighbor search is performed over each frame. The action is then recognized by taking the majority vote on the nearest neighbors found from the database.

2.1 Manual Feature Selection

The feature set is manually picked after visualizing the importance of different features to each action through 3D trajectory plots (Fig. 3). The features that have been tested on include position, velocity, acceleration, hand orientation and angular velocity.

Limitations

There are thousands of features possible; however, only 3 features can be plotted and visualized at a time. This approach is tedious and does not result in an optimal feature set.

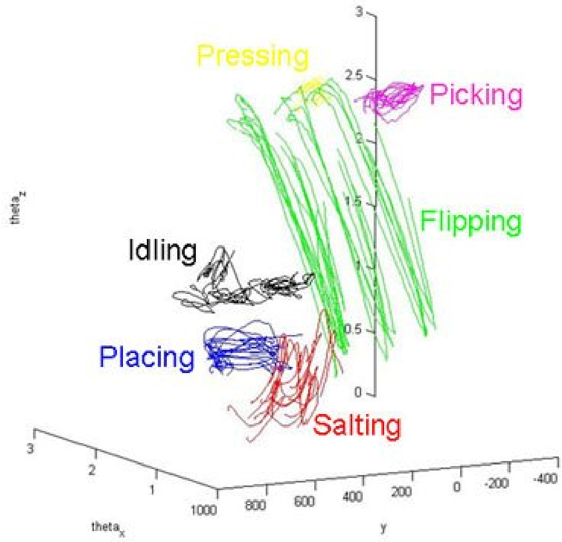

Figure 3: Visualizing features of different actions (position in y direction and orientations in x and z directions)

2.2 Results

The following are the results of using different window sizes, combinations of feature sets, and number of nearest neighbors:

Window size	Recognition rate (%)					
	Placing	Flipping	Salting	Pressing	Picking	*Average*
1	71.32	85.22	94.01	68.74	52.94	74.446
11	74.48	89.83	95.86	70.39	51.23	76.358
21	**74.46**	**90.54**	**96.3**	**71.16**	**57.72**	**78.036**
31	71.34	91.88	95.62	69.81	51.38	76.006
51	64.19	92.82	94.74	64.71	48.28	72.948

Feature set	Recognition rate (%)					
	Placing	Flipping	Salting	Pressing	Picking	*Average*
theta_x, y, theta_z - single	74.46	90.54	96.3	71.16	57.72	78.036
theta_x, y, theta_y, theta_z - single	80.72	89.3	96.3	75.75	57.24	79.862
y, theta_y, theta_z - both	86.33	84.17	96.3	85.37	57.24	81.882
theta_x, y, theta_y, theta_z, vz - both	85.67	85.01	97.31	88.81	71.97	85.754
theta_x/y/z, vx/y/z - both	85.83	88.45	97.31	86.8	68.17	85.312
theta_x/y/z, vx/y/z, y - both	**86.99**	**83.04**	**96.75**	**86.8**	**85.27**	**87.77**

Number of nearest neighbors	Recognition rate (%)					
	Placing	**Flipping**	**Salting**	**Pressing**	**Picking**	***Average***
1	85.67	85.01	97.31	88.81	71.97	85.754
3	85.83	84.9	97.42	87.95	75.53	86.326
5	86.33	85.07	97.19	86.51	78.86	86.792
7	**86.33**	**84.9**	**97.08**	**86.23**	**80.52**	**87.012**
9	86	85.35	96.63	85.8	80.52	86.86
15	86.66	84.56	95.51	83.21	80.05	85.998
40	86.99	84.51	95.06	81.21	80.05	85.564

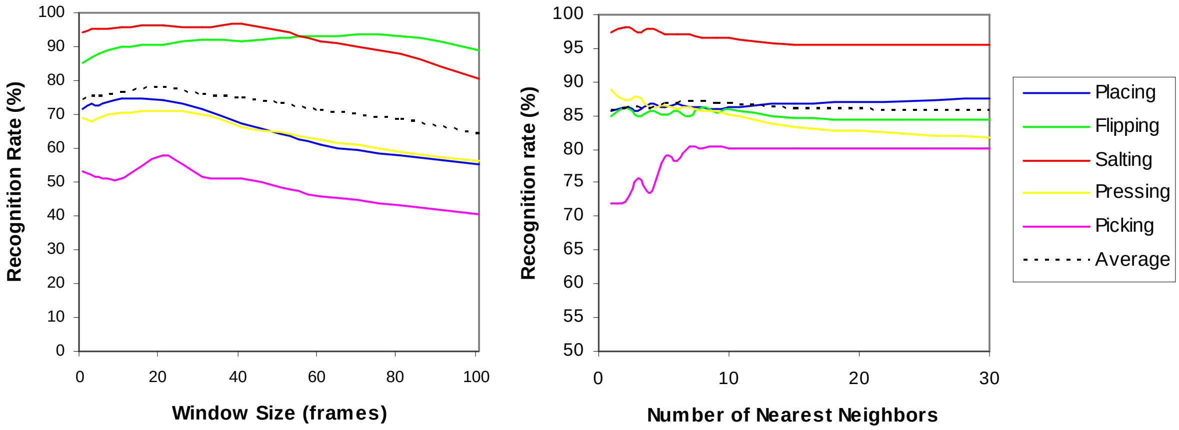

Figure 4: Experimental results for determining the best window size (left) and number of nearest neighbors (right).

3. Future Work

3.1 Automatic Feature Selection

In order to further improve the recognition rate, boosting techniques such as AdaBoost, an algorithm that combines weak classifiers to improve their performance by choosing appropriate weights and emphasizing previously misclassified data, will be implemented.

3.2 Accelerometers

Instead of using marker information from motion capturing for motion recognition, accelerometers can also be used to collect user motion data (Fig. 5).

Figure 5: Motion capture with reflective markers and accelerometers.