Hand Posture Control of a Robotic Wheelchair Using a Leap Motion Sensor and Block Sparse Representation based Classification

Ali Boyali*, Naohisa Hashimoto†, Osamu Matsumoto‡
National Institute of Advanced Industrial Science and Technology
Intelligent Systems Research Institute
Tsukuba, Ibaraki, Japan
Email:*ali-boyari@aist.go.jp, †naohisa-hashimoto@aist.go.jp, ‡matsumoto.o@aist.go.jp

***Abstract*—In this study, a gesture and posture recognition method which is based on the Block Sparse, Sparse Representative Classification, and its use for a robotic wheel-chair control are explained. A Leap Motion sensor is used to capture the postures of the left hand. There are five postures mapped to the control commands of the power wheel-chair. These commands can be expanded as the posture recognition commands can deal with high number of classes. The MATLAB functions used in the computations are compiled into .NET programing environment. We tested the hand posture control in a hall where are occupied by tables and chairs. The navigation experiments were successful.**

Keywords*—Robotic Wheel-chair control, Gesture and Posture Recognition, Compressed Sensing, Block Sparse Recovery, Sparse Representation based Classification.

I. INTRODUCTION

The trends in computing technologies are evolving towards the systems and algorithms that can sense the environment and make smart decisions to provide the users with more intuitive user interfaces. These smart interfaces pave the way for Human Machine Interfaces (HMIs) that aim to decrease the physical and cognitive loads of the users. As the machines and computing technologies get more compact and include variety of tiny sensors, their use by ordinary people becomes more difficult with the increasing functionality. Because of these developments, we need more intuitive and easy to use interfaces. These complex systems can be difficult even for ordinary users, but even more so when elderly or people who have some form of disability are concerned.

In this study, we explain a novel gesture and posture recognition algorithm and its deployment on a robotic wheelchair as a replacement of a conventional joystick control. The postures of left hand are captured via a Leap Motion sensor which can report the palm position, velocity and orientation values with the sub-millimeter accuracies.

The gesture and posture recognition algorithm is an implementation of the well-known Sparse Representation based Classification (SRC) algorithm [1] which has proven to be robust and highly accurate method in the case of face recognition research [1–3]. Based on this approach, Boyali et al. [4][5] extended the use of the SRC algorithm for gesture recognition and reported high classification accuracies. We further extend the SRC based gesture and posture recognition algorithm proposed by Boyali et al. in [4][5] by incorporating the Block and Group Sparsity approach which enhances the recognition accuracy as well as the speed of the algorithms.

The robotic wheelchair, on which the SRC based gesture and posture recognition algorithms are tested, is equipped with obstacle avoidance and path following systems which make the wheelchair fully autonomous [6].

This is an initial study of an end-user power wheel-chair control interface which will allow the people with severe mobility impairment to command the wheelchair with their residual and voluntary muscle movements. We employed a Leap Motion sensor to capture the postures of a hand to test the SRC based gesture and posture recognition algorithm, as the Leap Motion sensor has recently been introduced to the market. In future studies, we will utilize other means of sensors and systems with the introduced gesture and posture recognition algorithm.

Gesture or posture recognition based control of a power wheelchair has long been investigated by the researchers. There are several studies on the more intuitive user interfaces, to enable the people with severe mobility impairment and restricted muscle movements to command a power wheelchair. That is because the conventional joystick control of power wheelchairs may not be operated by those who are quadriplegics, handicapped children or people suffering from progressive Parkinson disease [7].

The remainder of the paper is organized as follows. In Section II, a brief literature review of alternative control interfaces for power wheelchair is given. The proposed BS-SRC based gesture recognition method is explained in Section III. Section IV is dedicated to the system and software architecture. In Section V, the implementation and simulations are detailed. The paper ends with the Conclusion Section.

II. PREVIOUS STUDIES

The alternative methods to conventional joystick wheelchair control are cluttered around different modalities. The most common modalities seen in the related literature are speech, gesture, gaze recognition and bio-signal control.

The voice and speech recognition systems suffer from the ambient noise and the accuracy rate depends on speech dexterity and pronunciation [8]. In the bio-signal based studies, Electromyography (EMG) [9–11], Electroencephalography (EEG) [12–14], Electrooculography (EOG) [15][16] are utilized to capture the occupant's intention for the power wheelchair control. In addition to the long bio-signal capturing time requirement and slow response, these studies also report more than 8-20 % mis-classification rates that render the proposed systems less appropriate for the real-time applications.

The gesture and posture recognition methods proposed by Jia et al. [7] and Rofer et al. [17] are based on the template matching and Finite State Machine methods. In the study by Jia et al. [7], the head of the occupant is monitored by a camera and the position of the head and possible head gestures are evaluated to find out whether the occupant intents to give a navigation command to the wheel-chair. The accuracy of the camera based studies highly depends on the varying illumination, indoor and outdoor environments, cluttered background and shadow. The studies by Rofer et al. [17] and Teymorian et al. [18] utilize the Finite State Machine method which relies on the experimentally defined threshold and dead-zone values. The state transitions are based on these manually defined values which limit the free motion of the tracked body parts. The bio-signal and head-joystick based systems require attached or worn devices which may disturb the wheel-chair occupant.

The proposed approach in this study is easy to implement, highly accurate and does not bring about any limitation to the free motion of the tracked hand. It is also robust to the illumination variations and there is almost no mis-classification error.

III.Block Sparse, Sparse Representation based Classification

The BS-SRC algorithm is based on the Sparse Representative based Classification algorithm which was originally implemented for face recognition [1]. The idea is simple. For a given collection of samples, which are stacked as a column vector in a matrix $A = [A_1, A_2, \ldots, A_n] = [v_{1,1}, v_{1,2} \ldots, v_{k,n_k}] \in \mathbb{R}^{mxn}$, the observed pattern $y \in \mathbb{R}^m$ is assumed to be represented by the linear combination of the samples stacked in the dictionary matrix. Accordingly, the vector $x_0 = [0, \ldots, 0, \alpha_1, \alpha_1, \ldots, \alpha_k, 0, \ldots, 0] \in \mathbb{R}^n$ which represents the linear span coefficients are obtained by solving the equation $y = Ax + v$ where $v \sim \mathcal{N}(0, \sigma^2)$ is the white noise. In the solution, the compressed sensing principles and ℓ^1 minimization methods are utilized. Once the solution vector is recovered by using a few linear measurements, the class label r_i is defined by finding the minimum reconstruction residual using equation (1).

$$\min_i \quad r_i(y) = \|y - A\delta_i(\hat{x_1})\|_2 \tag{1}$$

Boyali et al. [4][5] adopted the SRC method for the gesture recognition and reported very high accuracy rates. The authors take the Discrete Cosine Transform (DCT) of the two dimensional gestures which consist of only x and y coordinates to construct the dictionary matrix.

The recently introduced and highly effective approach in compressed sensing literature is the block or group structure assumption in seeking a sparse solution [19–21]. In this approach, the sparse solution has a structure in which the elements of the groups become either collectively zero or take non-zero values.

Several solutions have been proposed for the block sparse problems, such as Block Sparse Bayesian Learning (BSBL) [22], Dynamic Group Sparsity (DGS) [19] and Block Sparse Convex Programming (BS-CP) [23] which exploit the group sparsity of the solution.

In the SRC method, the observed pattern is assumed to be represented by all the samples stacked in the dictionary matrix and the coefficients of the unrelated class samples are also computed during the optimization. The block sparsity approach eliminates these extra computations, thus yielding a faster solution.

When the SRC method is concerned, there is indeed a structure in the sparse solution x, since the samples from the classes are put in the dictionary matrix in a structured form. The sparse solution, when obtained via Block Sparsity methods, contains the coefficients belonging to the corresponding class only.

We made use of the three methods BSBL, DGS and BS-CP for two different gesture sets given in the studies by Boyali et al. [4][5]. The experiments and simulations gave more accurate results than those reported in the previous studies in which only the SRC method is used. Besides, the block sparsity approaches make the algorithms faster than the classical sparsity assumptions, making the algorithms good candidates for real time implementations.

The BSBL approach is chosen for the posture recognition and its real-time implementation in this study. There are two algorithms proposed in the BSBL method, the BSBL-EM in which the optimization problem is solved by Expectation Maximization (EM), and the BSBL-BO which utilizes the Bound Optimization method for the solution. The latter one is faster then the former.

In the BSBL method, the sparse solution:

$$x = [\underbrace{x_1, \ldots, x_{d1}}_{x_1{}^T}, \ldots, \underbrace{x_{d_{g-1}+1}, \ldots, x_{d_g}}_{x_g{}^T}] \in \mathbb{R}^n \tag{2}$$

consists of concatenated g blocks and each block, $x_i \in \mathbb{R}^{d_i x 1}$ is assumed to be generated by a parametrized multivariate Gaussian distribution:

$$p(x_i; \gamma_i, B_i) \sim \mathcal{N}(0, \gamma_i B_i) \quad i = 1, \ldots, g$$

with a non-negative parameter γ_i which controls the block sparsity of x and a positive definite correlation matrix B_i, which maintains the correlation structure of the i^{th} block.

The parameters are estimated by a type-II maximum likelihood procedure [24] after the parameters, γ_i, B_i and the standard deviation of the measurement noise λ, the posterior mean and covariance matrices are updated by using the EM or BO methods.

IV. System Architecture

The robotic wheel-chair used in the study was previously designed at The National Institute of Advanced Industrial Science and Technology (AIST) laboratories (Fig. 1). It has an autonomous mode, by which the wheel-chair can travel between two pre-defined points [6]. In this mode, since the occupant's hands become free, the hand postures and gestures can be utilized to add additional functionalities to the system. We use free motion and postures of a hand captured by an integrated Leap Motion sensor to command the wheelchair. An additional modality also improves the safety while traveling. In this configuration, the Leap Motion sensor is located on the left side of the wheel-chair.

 ISBN: 978-1-61208-363-6

Figure 1. Robotic Wheel-Chair and Leap Motion Sensor

There are five navigation commands, including, but not limited to, "Go Straight, Turn Left, Turn Right, Stop, and Reverse" on the wheel-chair navigation mode (Fig. 2). The algorithm has no limitations for adding more posture classes into the dictionary matrix, and the number of the posture classes can be increased when necessary.

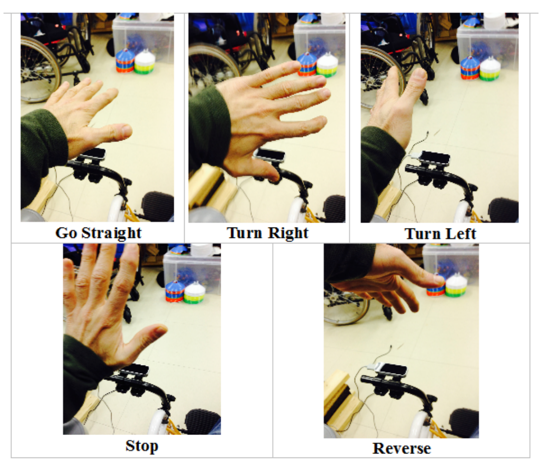

Figure 2. Hand Postures Mapped to Wheel-chair Navigation Commands

The posture recognition module is located on a different computer and the recognized commands are sent to the main computer on which the obstacle avoidance, and autonomous navigation modules reside, via a network cable (Fig. 3).

The Leap Motion sensor is a new device consisting of infra-

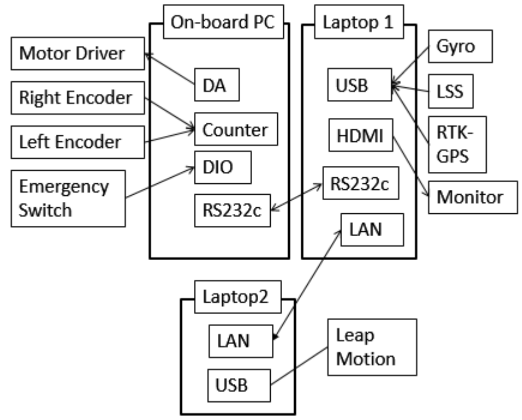

Figure 3. System Architecture

red cameras (Fig. 4) which can detect hands or pointers and reports the position, orientation and velocity of the tracked object at a frequency around 80 Hertz. The effective tracking volume is of a pyramid form, the height of which is about 47 cm from the center of the leap motion to the base of the pyramid.

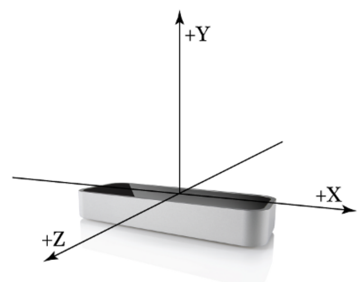

Figure 4. Leap Motion Sensor and Its Coordinate System

The angles of view along the z and x axes are 2 and 2.3 radians, respectively. The tracking volume is sufficient for this study. As the left hand is supported by the wheel-chair arm, it is always visible to the system when a posture based mode is active. The posture control mode is activated when the Leap Motion sensor detects a hand in the viewing volume. If there is no visible hand, the system enters the autonomous or joystick mode.

Although the Leap Motion sensor provides many variables, such as the number of fingers, finger direction, their relative positions to each other, finger and palm velocities as well as some internal recognized gestures by the frame object, we only utilized the palm orientations; roll, pitch and yaw angles, the direction of the normal vector of the palm and the palm velocities for posture recognition.

The MatLeap MATLAB Mex interface [25] is used to analyze the Leap Motion data in MATLAB environment, as the proposed BS-SRC based gesture recognition algorithm written in MATLAB. The MatLeap Mex interface contains of only a few functions and receives a few variables from a Leap Motion sensor. We added functions to the MatLeap interface to read the variables related to the tracked palm. The detected

 ISBN: 978-1-61208-363-6

postures of the hand are then used by the BS-SRC algorithm for classification.

The algorithms and codes are converted to a dll file using the MATLAB's deployment toolbox to be able to use the available solvers and the implemented codes for posture recognition in the CSharp programming environment.

V. Implementation and Simulations

When the gesture and postures are mapped to a certain number of states, the common method is to use Finite State Machines (FSM), if the number of features is limited to a few dead-zone or threshold values which do not limit the freedom. Using the FSM approach may be a valid decision when the boundaries can be strictly separable by deadzones. However, the uncertainties naturally exist in the complex signals such as bio-signals which are required to be collected for relatively long periods of time. Similarly, as the Leap Motion sensor reports the features at a high frequency, and posture and gesture recognition requires a certain number of data samples, the boundaries between the postures cannot be set by FSM methods in a feasible way.

The BS-SRC based gesture recognition method yields highly accurate results which are more than 99% for different kinds of gesture sets. We tested the BS-SRC method for posture recognition and obtained more accurate results than the gesture recognition studies.

The first step in posture recognition is to construct a dictionary matrix $A = [A_1, A_2, \ldots, A_n] = [v_{1,1}, v_{1,2} \ldots, v_{k,n_k}]$ in which the samples from five postures are stacked as the column vectors. The DCT coefficients of the matrix are then computed.

Each posture signal vector v_{k,n_k} consists of the palm roll, pitch and yaw angles and the direction of the palm normal vector in the three axes x, y and z. The left hand is kept at the specified posture for 30 seconds and the palm roll, pitch and yaw, as well as the palm normal vector direction values are recorded. We reduced the sampling frequency from 80 Hertz to 50 Hertz. Each posture sample has a length of a half second, which corresponds to 25 measured values from each feature. There are 30 snapshots of each postures in the dictionary matrix $A \in \mathbb{R}^{150x150}$.

These sample postures are put in the dictionary matrix, the rest of 30 posture samples from each class are used for verification. In total there are 150 test postures and the BS-SRC based algorithm yields 100% recognition accuracy for the test samples (Fig. 5).

In the real-time implementation of the posture recognition algorithm, there are transition states between the postures. These transition states which can be considered as gesture, have a very short duration when compared to the postural states. Each transition states or gestures differ in length. During the transition, the hand posture can only be either of one of the involved states, and the algorithm chooses one of them.

Only a few false spikes occurred in the simulations. As shown in Fig. 6, the algorithm gives two instances of false recognition, while the hand repeatedly switchs between the Go Straight and Turn Right states.

We eliminate these rarely seen false recognitions by applying a very simple signal filtering method which incurs a short

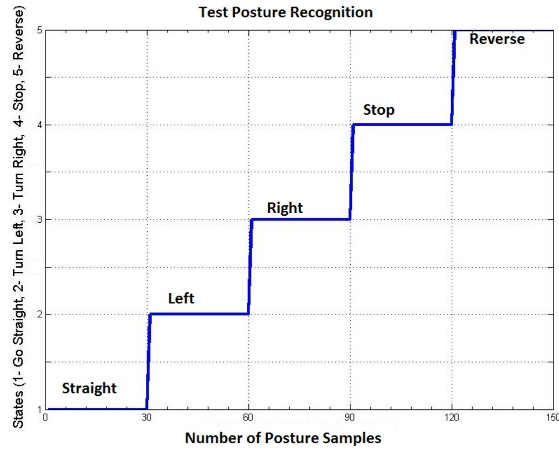

Figure 5. Test Postures and Recognized States

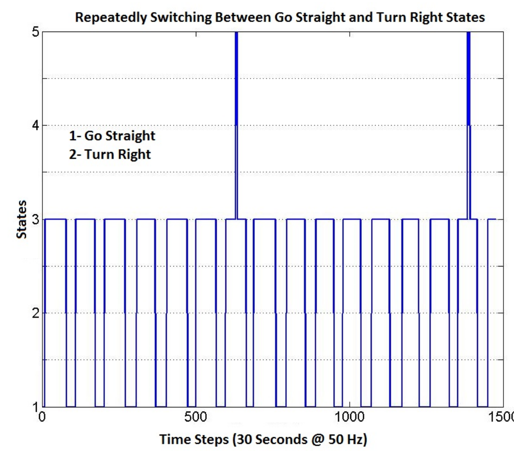

Figure 6. Switching Between Turn Right and Go Straight Postures

delay to the system. The delay is acceptable to make sure we produce a robust implementation of the algorithm. In most cases, there is no need to use such a filter. The figures (Figures 7 and 8) show the unfiltered simulation results. We can also use a low rank matrix recovery method to detect if the measured instance is a posture or gesture. For this approach, in every time instance, we add the received values to the dictionary matrix as the 151^{th} column and separate the matrix into two components; low rank dictionary and the outliers matrix. We use Direct Robust Matrix Factorization (DRMF) [26] algorithm to detect whether the current measurement is a posture or a gesture. Since the gesture states are not represented in the posture dictionary, the DRMF algorithm gives higher residual error for these states. There is an alternative approach we have been working on. In this approach, once the transition states are detected by the low rank matrix factorization method, we can employ a second dictionary which only consists of the gesture classes. The difficult part of this approach is to spot exemplar gestures from the streaming signals.

The algorithm buffers 25 samples from each of the features, concatenates them in a column vector to build the dictionary matrix. The recognized postures are numbered one to five and sent to the Laptop-1 (Fig. 3) computer via a network cable.

 ISBN: 978-1-61208-363-6

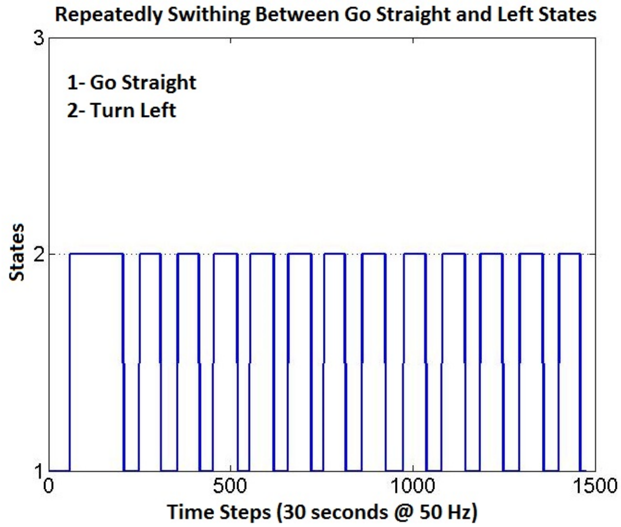

Figure 7. Switching Between Turn Left and Go Straight Postures

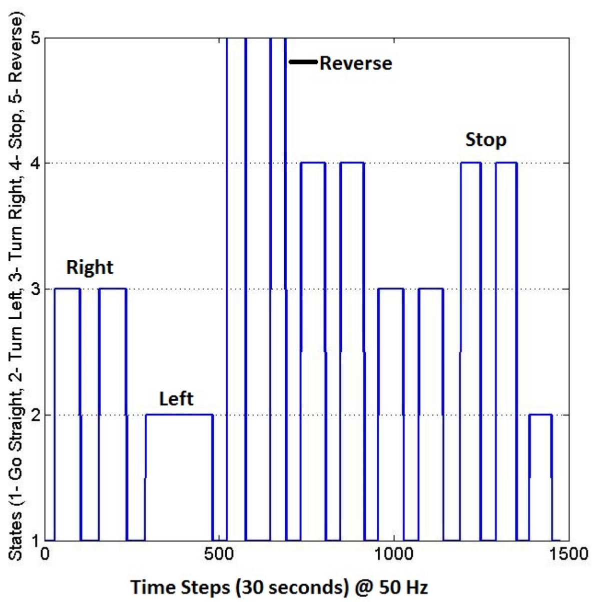

Figure 8. All States Visited

The receiving computer sends commands to the on-board PC to control the electric motors for a given navigation method. We tested the real-time performance of the power wheelchair. The tests were performed in a hall where there were tables and chairs and we navigated in this crowded area by controlling the wheelchair hand postures only.

VI. Conclusion and Future Works

This study is aimed to test the real-time performance of the BS-SRC gesture and posture recognition method on a power-wheelchair. The hand postures are captured via a Leap Motion sensor by collecting the palm roll, pitch and yaw angles as well as three palm normal direction vector at each time instance, then the concatenated features are evaluated by the BS-SRC method. The method yield highly accurate and robust

Figure 9. Test Hall

recognition rates.

The BS-SRC method is tested originally for two different gesture sets and we received higher accuracies and faster recognition when compared to those of the studies [4][5] in which only the SRC method was used. We will design an additional dictionary matrix, the column vectors of which will only consist of the gestures. Since the Leap Motion reports observations in a streaming signal form, the difficulty in this approach is to spot the beginning and end points of the gestures for a proper construction of the dictionary matrix.

We tested the posture recognition algorithm for only the Leap Motion sensor. We aim to repeat the experiments by using a pressure distribution sensor by which we can detect the seating postures, to track the change of weight center of the occupant and Microsoft's MYO gesture bracelets which is still in development phase and is not introduced to the market yet.

Acknowledgment

This study is supported by the Japan Society for the Promotion of Science fellowship program.

References

[1] J. Wright, A. Y. Yang, A. Ganesh, S. S. Shankar, and Y. Ma, "Robust face recognition via sparse representation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 2, 2009, pp. 210–227, ISSN:0162-8828.

[2] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. S. Huang, and S. Yan, "Sparse representation for computer vision and pattern recognition," Proceedings of the IEEE, vol. 98, no. 6, 2010, pp. 1031–1044.

[3] T. Li and Z. Zhang, "Robust face recognition via block sparse bayesian learning," Mathematical Problems in Engineering, vol. 2013, 2013.

[4] A. Boyali and M. Kavakli, "A robust gesture recognition algorithm based on sparse representation, random projections and compressed sensing," in Proceedings of the 7^{th} International Conference on Industrial Electronics and Applications (ICIEA 2012) July 18–20, 2012, Singapore. IEEE, pp. 290–294.

 ISBN: 978-1-61208-363-6

[5] A. Boyali and M. Kavakli, “A robust and fast gesture recognition method for wearable sensing garments,” in Proceedings of the 4^{th} International Conferences on Advances in Multimedia (MMEDIA 2012) Apr 29–May 4, 2012, Chamonix / Mont Blanc, France. IEEE, pp. 142–147.

[6] N. Hashimoto, U. Ozguner, N. Sawant, K. S. Yokuzuka, M, O. Matsumoto, and S. Tsukuwa, “A framework of fault detection algorithm for intelligent vehicle by using real experimental data,” in Proceedings of the 14^{th} International IEEE Conference on Intelligent Transportation Systems (ITSC 2011) Oct 5–7, 2011, Washington, DC, USA. IEEE, pp. 913–918.

[7] P. Jia, H. H. Hu, T. Lu, and K. Yuan, “Head gesture recognition for hands-free control of an intelligent wheelchair,” Industrial Robot: An International Journal, vol. 34, no. 1, 2007, pp. 60–68.

[8] O. Babri, S. Malik, T. Ibrahim, and Z. Ahmed, “Voice controlled motorized wheelchair with real time obstacle avoidance,” in Proceedings of the 3^{rd} International Conference on Communications and Information Technology (ICCIT 2012), Beirut, Lebanon. IEEE.

[9] T. Felzer and B. Freisleben, “Hawcos: the hands-free wheelchair control system,” in Proceedings of the fifth international ACM conference on Assistive technologies, ACM. ACM, pp. 127–134.

[10] C. S. L. Tsui, P. Jia, J. Q. Gan, H. Hu, and K. Yuan, “Emg-based hands-free wheelchair control with eog attention shift detection,” in IEEE International Conference on Robotics and Biomimetics, ROBIO 2007, Dec 15–18, 2007, Sanya, China, IEEE. IEEE, pp. 1266–1271.

[11] Y. Oonishi, S. Oh, and Y. Hori, “New control method for power-assisted wheelchair based on upper extremity movement using surface myoelectric signal,” in Proceedings of the 10^{th} IEEE International Workshop on Advanced Motion Control, 2008, (AMC’08), Mar, 26–28, Trento, Italy, IEEE. IEEE, pp. 498–503.

[12] D. A. Craig and H. Nguyen, “Adaptive eeg thought pattern classifier for advanced wheelchair control,” in Proceedings of the 29^{th} Annual International Conference of the IEEE, Engineering in Medicine and Biology Society, (EMBS 2007), Aug 23–26, 2007, Lyon, France, IEEE. IEEE, pp. 2544–2547.

[13] I. Iturrate, J. Antelis, and J. Minguez, “Synchronous eeg brain-actuated wheelchair with automated navigation,” in Proceedings of the IEEE International Conference on Robotics and Automation, 2009. (ICRA’09), May 12–17, Kobe, Japan, IEEE. IEEE, pp. 2318–2325.

[14] R. Gopura, D. Bandara, J. Gunasekara, and T. Jayawardane, “Recent trends in emg-based control methods for assistive robots,” Electrodiagnosis in New Frontiers of Clinical Research (Turker, H., Hg.), S, 2013, pp. 237–268.

[15] R. Barea, L. Boquete, M. Mazo, and E. L. A. García-Lledó, “Eog technique to guide a wheelchair,” in Proceedings of the 15^{th} International Conference of Pattern Recognition Sept. 03–07, 2000, IEEE. IEEE, pp. 668–671.

[16] R. Barea, L. Boquete, M. Mazo, and E. López, “Wheelchair guidance strategies using eog,” Journal of Intelligent and Robotic Systems, vol. 34, no. 3, 2002, pp. 279–299.

[17] T. Rofer, C. Mandel, and T. Laue, “Controlling an automated wheelchair via joystick/head-joystick supported by smart driving assistance,” in Proceedings of the 11^{th} IEEE International Conference on Rehabilitation Robotics, (ICORR 2009), June 23–26, 2009, Kyoto, Japan. IEEE, pp. 743–748.

[18] A. Teymourian, T. Lüth, A. Graeser, T. Felzer, and R. Nordmann, “Brain-controlled finite state machine for wheelchair navigation,” in Proceedings of the 10^{th} International ACM SIGACCESS conference on Computers and accessibility, Oct, 13–15, 2008, Scotia, Canada. ACM, pp. 257–258.

[19] J. Huang, T. Zhang, and D. Metaxas, “Learning with structured sparsity,” The Journal of Machine Learning Research, vol. 12, 2011, pp. 3371–3412.

[20] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, “Model-based compressive sensing,” IEEE Transactions on Information Theory, vol. 56, no. 4, 2010, pp. 1982–2001.

[21] Y. C. Eldar, P. Kuppinger, and H. Bolcskei, “Block-sparse signals: Uncertainty relations and efficient recovery,” IEEE Transactions on Signal Processing, vol. 58, no. 6, 2010, pp. 3042–3054.

[22] Z. Zhang and B. D. Rao, “Recovery of block sparse signals using the framework of block sparse bayesian learning,” in Proceedings of the 37^{th} IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), March, 25–30, 2012, Kyoto, Japan. IEEE, pp. 3345–3348.

[23] E. Elhamifar and R. Vidal, “Block-sparse recovery via convex optimization,” IEEE Transactions on Signal Processing, vol. 60, no. 8, 2012, pp. 4094–4107.

[24] M. E. Tipping, “Sparse bayesian learning and the relevance vector machine,” The journal of machine learning research, vol. 1, 2001, pp. 211–244.

[25] “Matleap mex interface for the leap motion sensor,” 2014, URL: https://github.com/jeffsp/matleap [accessed: 2014-02-28].

[26] L. Xiong, X. Chen, and J. Schneider, “Direct robust matrix factorizatoin for anomaly detection,” in Proceedings of the 11^{th} International Conference on the Data Mining (ICDM), Dec. 11-14, 2011, Vancouver. IEEE, pp. 844–853.

 ISBN: 978-1-61208-363-6