Proceedings of the 2nd World Congress on Electrical Engineering and Computer Systems and Science (EECSS'16)
Budapest, Hungary – August 16 – 17, 2016
Paper No. MVML 110
DOI: 10.11159/mvml16.110

Human Motion Detection in Manufacturing Process

Ágnes Lipovits, Mónika Gál, Péter József Kiss, Csaba Süveges
University of Pannonia
Egyetem str. 10., Veszprém, Hungary
lipovitsa@almos.uni-pannon.hu; galbmonika@almos.uni-pannon.hu; kiss.peterjozsef@virt.uni-pannon.hu

Abstract - In this paper we present a new motion recognition algorithm based on skeleton and depth map data extraction from two generations of the Kinect sensors as a first step to support manufacturing process optimization. A real production line activity was simulated in a laboratory environment and a specific motion - the barcode scanning - was recognized and validated by semi-automated log file processing. We show that the proposed methods give appropriate accuracy by using whichever Kinect sensor.

***Keywords*:** human motion detection, Kinect, skeleton, depth map

1. Introduction

Optimization of production lines can increase the efficiency of the manufacturing system. To find an effective solution to this problem optimization algorithms need a lot of input information. The main goal of our research is to automatically ensure sufficient information for the product line optimization methods in a computer assembly factory.

Several researches were carried out in the 80's [1] showing the importance of the ergonomics of the assembly workstations. Although the aim is to fully automatize manufacturing processes, in some cases the production line employee could not be replaced with robotic techniques, by the reason of the frequent changeovers, the new and small amount products. Moreover, some production processes are too complex and difficult to be automatized by current technology. So the ergonomic and fast work of the employee is required for the optimal production of the factory and should be taken into account at the optimization method as an important factor.

One possibility to get sufficient information about the time requirement of a workstation is following the products. This method does not provide any information about the effective working time and the time requirement, it only shows the actual throughput capacity. Another possibility is the recognition of gestures and motion of workers. In this article we introduce the first step of our research, a method for cycle time estimation in industrial environment based on skeleton and depth data of a Kinect sensor that are analyzed with data mining methods. Another contribution of our work is showing the efficiency differences between the different generations of Kinect sensors in such an application.

Gesture recognition is an active research area in the field of computer vision. The goal is often to understand only hand gestures [2][3], in other cases we would like to recognize whole body gestures [5][4]. Initial data used for the extraction of high-level information is divided into three main groups: depth-map [6], 3D point cloud [2] and skeleton information [5][4].

In [2] the first step is to find one point of the hand using a 3D hand tracking algorithm. Then the hand region is segmented out from depth image and is converted to a point cloud. For the segmentation they use a 1-d connect component selection algorithm. From the point cloud information 3D moment invariants are computed. Finally, the Support Vector Machine (SVM) is applied for the classification.

Eight hand and head gestures are distinguished in [6] using solely depth information. The first step of this method is segmentation aiming at separating the human body and the background. This is done by using auto thresholding on the depth histogram. The threshold is found from the valley following the first large peak of the histogram. The foreground depth image is divided into 14x14 elements and each grid element is parameterized with depth change and motion information. The depth information for a grid element is divided into not equal parts and the corresponding pixels are counted and normalized. The motion information for a grid element is also calculated from the depth image. The successive depth images are subtracted from each other, next noise reduction is carried out and then the result is converted

to a binary image, where white pixels represent the motion content. White pixels in a grid element are normalized. Multiclass SVM is used in training the 8 gesture-classes.

[5] deals with challenging tasks, such as the real-time implementation of a gesture recognition system, the temporal resolution of gestures and the task of creating a user independent gesture classifier. They compute node angles and node quaternion from the initial skeleton information, and then 10 different right hand gestures are distinguished by a neural network using positive and negative examples. A backpropagation algorithm is used for the training phase, as the best configuration the hidden nodes are selected. The gesture lengths are dependent on the person, so they apply person re-identification module based on mainly silhouette segmentation and a connected graph in which the nodes contain the color histogram of the region and the edges containing the neighborhood information. Furthermore, they apply the Fast Fourier Transformation (FFT) and tacking the position of the fundamental harmonic component, the period could be evaluated as the reciprocal value of the peak position, so the gesture lengths are estimated.

In [4] 6 body gestures are distinguished based on skeleton information. The extracted feature vector contains the joint coordinates, 3 xyz-velocities per joints, 35 joint angles and 35 joint angular velocities. Besides extracting angles between adjacent segments, they also place an imaginary joint to (0, 0, 0) in world coordinates, which is the location of the camera. The feature vectors are classified using three methods: Naive Bayes, SVM and Random Forest.

In [7] the goal is to be able to distinct 10 gestures, where they combine the skeleton joint angles and the relative position of 6 arm joints and the head joint. K-means is used for clustering and the new gestures are classified using a Euclidean distance metric.

In [3] the goal is to recognize hand gestures and match them with different meanings (for example the V – means Victory or number two). So this task requires the identification of fingers. The first step is to detect the hand on the depth image, then a contour tracking algorithm detects the contours of the hand. The next step is the identification of fingers and a three-layer-classifier: finger counting, finger name collecting, and vector matching.
In the industrial environment the pose estimation for planning ergonomic motion is very frequent [8].

2. Related Techniques

2.1. Descriptors

In our method we use a depth map similar to [6] and a skeleton like in [4] [5] as input data. Our goal is to compare data deriving from different generations of Kinect sensors [9]. The extracted information is shown on fig. 1.

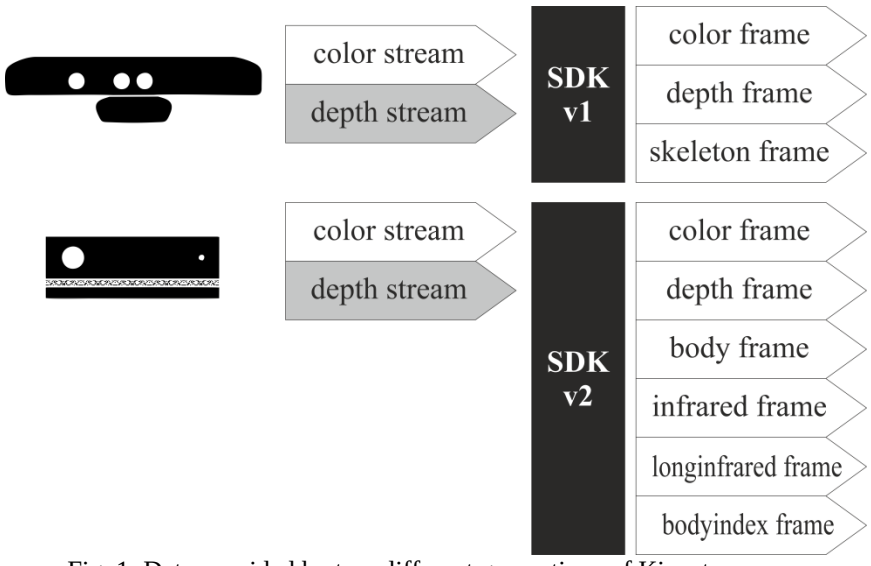

Fig. 1: Data provided by two different generations of Kinect sensors.

For the descriptor that is based on the depth map the first step is the human body segmentation (fig. 3). For this task we apply one of the Kinect SDK classes (named BodyIndexFrame) that represents a frame with depth or infrared pixels

belonging to tracked people. (This task in [6] was performed with histogram thresholding, but we have found it is less effective than BodyIndexFrame.) We perform histogram equalization on the depth image after the human body is segmented (fig. 3), so we get more detailed depth information.

Then the image without the background is divided into 196 blocks and every block is descripted by depth changing and motion information as follows.

In case of the depth information the histogram belonging to one of the blocks is divided into more different sized ranges (0-20, 21-30, 31-40, 41-50, 51-60, 61-80, 81-100, 101-120, 121-165, 166-255 and we count the pixels belonging to these ranges, then these values are normalized.

Motion information is also extracted from the depth image. The successive depth images are subtracted from each other. We perform noise reduction on the difference images and convert them to binary images on which the white pixels denote motion content. These binary images are also divided into 196 blocks, and we count the white pixels in each block then normalize these values.

Fig. 2: Depth image sequence after the human body segmentation (every 10th frame).

The length of the final depth image-descriptor is 2156. The depth information includes 196 blocks with 10 ranges-information and the motion information includes 196 blocks information.

In case of the other descriptor, which is based on the skeleton information, we also use temporal and spatial features as [4], but we calculated the displacement every second, based on a time stamp instead of calculating it for every 35 frames as in [4]. We have decided to do so because the Kinect sensor data frame rate is not consistent, it depends on how busy the sensor is. The 190 length feature vector is structured as follows: x,y,z coordinates of joints (60); x, y, z displacement per second (60); joint angles (17) (fig. 4); joint angles changes per second (17); joint-camera angles (18); joint-camera angles (fig. 4) changes per second (18).

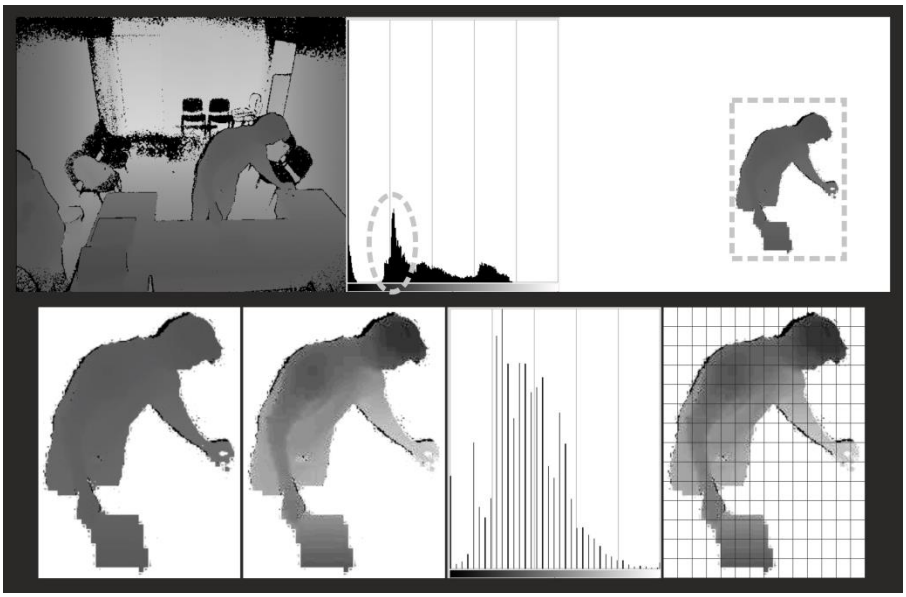

Fig. 3: Processing of a depth image: human body segmentation; histogram equalization of human body depth image; dividing it into 196 blocks.

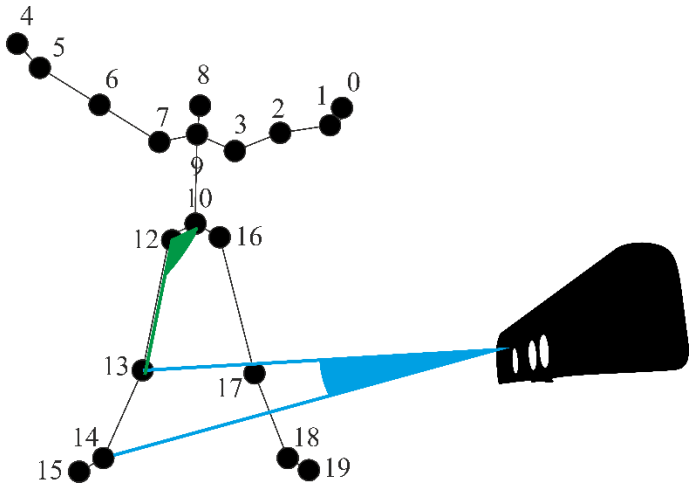

Fig. 4: Joint angles (green highlight); Joint camera angles (blue highlight).

2.2. Classification

Three different classification trees and their fusions were applied on descriptors extracted from skeletons and depth images (fig.5). Classification and Regression Trees (CRT) [10] [11] splits the data into sections that are as homogeneous as possible with respect to the dependent variable. Chi-squared Automatic Interaction Detection (CHAID)[12] selects the independent variable that has the strongest relation with the dependent variable at each step by using Person's Chi-square tests. Categories of each independent variable are merged if they are not significantly different with respect to the dependent variable. Quick, Unbiased, Efficient Statistical Tree (QUEST) [13] can use different statistic (F and χ^2statistic) for independent variable selection reducing the bias. All classification methods were validated by 10-fold crossvalidation. We applied Discriminant analysis [14] for the fusion of different classifiers' results using predicted values and predicted probabilities.

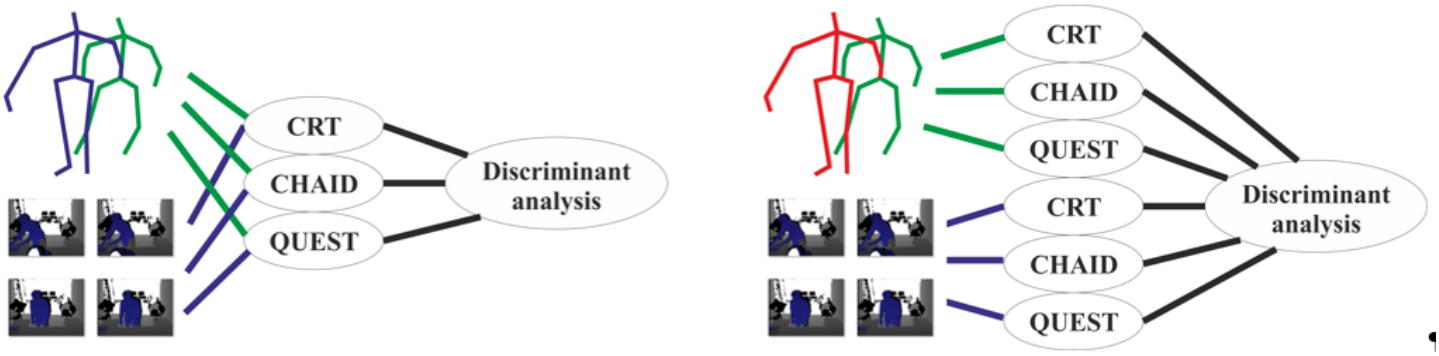

Fig. 5: Classification methods.

3. Experiment

In our lab we have artificially simulated a manufacturing environment. 6 people participated in the experiment, whom we observed with two different generations of Kinect sensors. The participants have tested 11 products. Their task was to investigate the products and look for any damages, then separate the damaged pieces from the intact items. As a first step in the product line optimization we would like to measure cycle times automatically based on information extracted from the Kinect sensor and using semi-automated validation produced by log process. The aim of our method is to recognize effectively the bar code scanning - one of the gesture of the product analysis. The cycle times are shown on fig. 6.

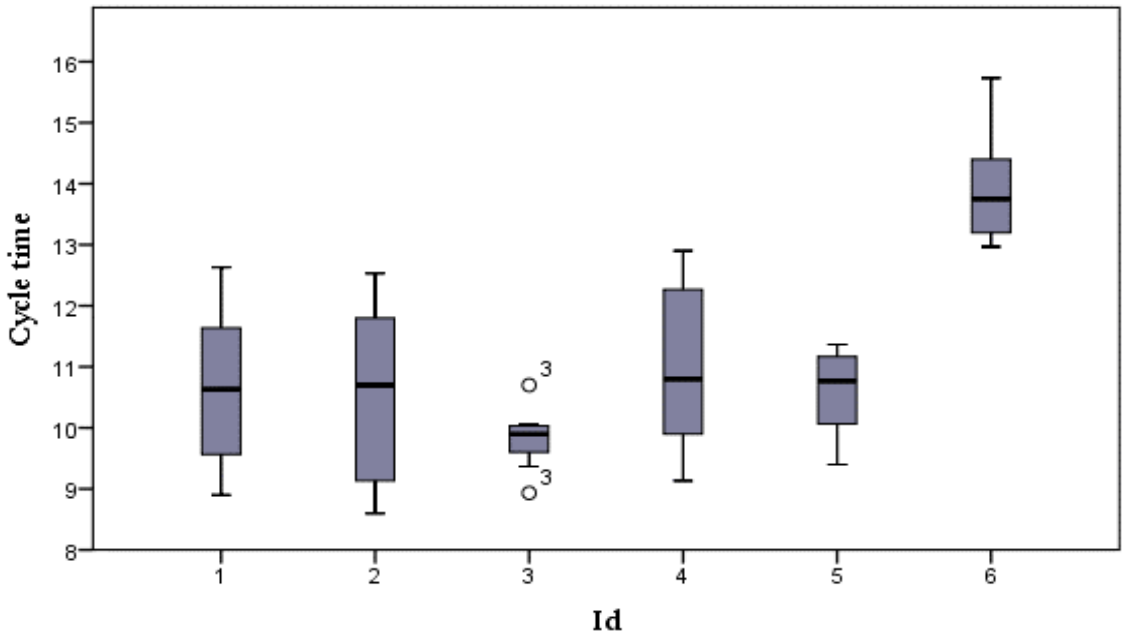

Fig. 6: Cycle times for 6 different people.

4. Results

The fused descriptor involved both skeleton and depth map and applied the three decision trees predictions give the best classification results (Overall Accuracy (OA) is 98.6 percent and F measure is 0.95) (Table 1). Both types of Kinect sensor can be used for this assignment, nevertheless the Kinect v2 provides better accuracy on average.

Table 1: Motion Recognition Results.

Descriptor	Method	Overall Accuracy (%)	F measure
Kinect v1 depth	CHAID	94.18	0.8059
Kinect v1 depth	CRT	95.14	0.8365
Kinect v1 depth	QUEST	93.78	0.7909
Kinect v1 depth	**late fusion**	96.50	0.8863
Kinect v1 skeleton	**CHAID**	95.67	0.8157
Kinect v1 skeleton	CRT	93.57	0.8540
Kinect v1 skeleton	QUEST	94.01	0.7830
Kinect v1 skeleton	**late fusion**	96.66	0.8864
Kinect v1 fusion	**CHAID**	97.57	0.9242
Kinect v1 fusion	CRT	97.05	0.9060
Kinect v1 fusion	QUEST	96.40	0.8827
Kinect v1 fusion	**late fusion**	**98.62**	**0.9558**
Kinect v2 depth	CHAID	95.92	0.8674
Kinect v2 depth	CRT	95.27	0.8494
Kinect v2 depth	**QUEST**	96.53	0.8850
Kinect v2 depth	late fusion	96.95	0.8882
Kinect v2 skeleton	**CHAID**	96.82	0.8914
Kinect v2 skeleton	CRT	95.88	0.8563
Kinect v2 skeleton	QUEST	96.73	0.8883
Kinect v2 skeleton	late fusion	97.07	0.8932
Kinect v2 fusion	**CHAID**	97.55	0.9160
Kinect v2 fusion	CRT	96.29	0.8687
Kinect v2 fusion	QUEST	97.03	0.8915
Kinect v2 fusion	**late fusion**	**98.73**	**0.9527**

The Receiver operating characteristic (ROC) curves (fig. 7) show the performance of the CHAID, QUEST and CRT classifiers.

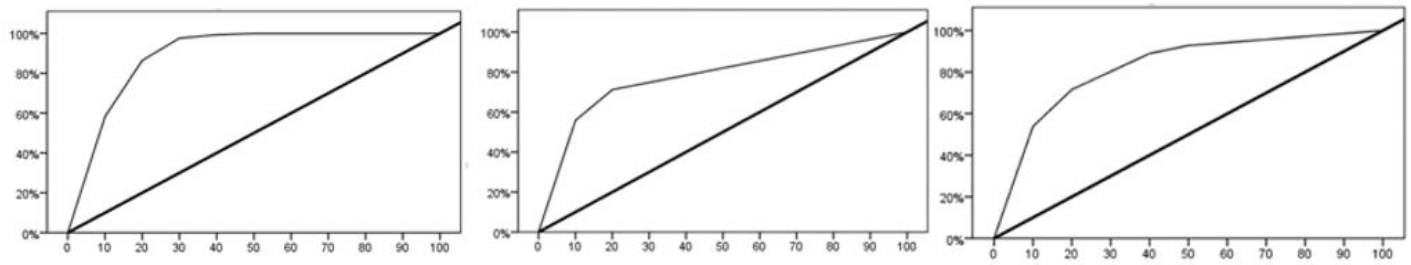

Fig. 7: ROC curves of CHAID (left), QUEST (middle), CRT (right) using late fusion of Kinect v1 data fusion.

The features obtained from skeleton data seem to be slightly more relevant then depth map descriptor with respect to not only the accuracy but also at the most important independent variables (fig. 7). Six pieces from skeleton descriptor (CamAng6_5 – means the camera angle between joint 6 and joint 5 (fig. 4.), HandRightVelocity_Z – means the z coordinate of the (HandRight) joint 4 displacement per second, WristRightVelocity_Z – means the z coordinate of the (WristRight) joint 3 displacement per second, CamAng5_4 and CamAng7_6 – means the camera angle between joint 5 and joint 4, and between joint 7 and joint 6) and four pieces from depth map descriptor (V1493, V1780, V1232, V1770) can be found among the 10 most important independent variables (Table 2).

Table 2: Importance and normalized importance of the 10 most important independent variables.

Independent Variable	Importance	Normalized Importance
CamAng6_5	0.128	100.0%
HandRightVelocity_Z	0.123	96.1%
WristRightVelocity_Z	0.104	81.3%
CamVel6_5	0.064	49.4%
V1493	0.060	47.0%
V1780	0.045	35.4%
CamAng5_4	0.045	34.8%
CamAng7_6	0.044	34.5%
V1232	0.043	33.3%
V1770	0.042	32.7%

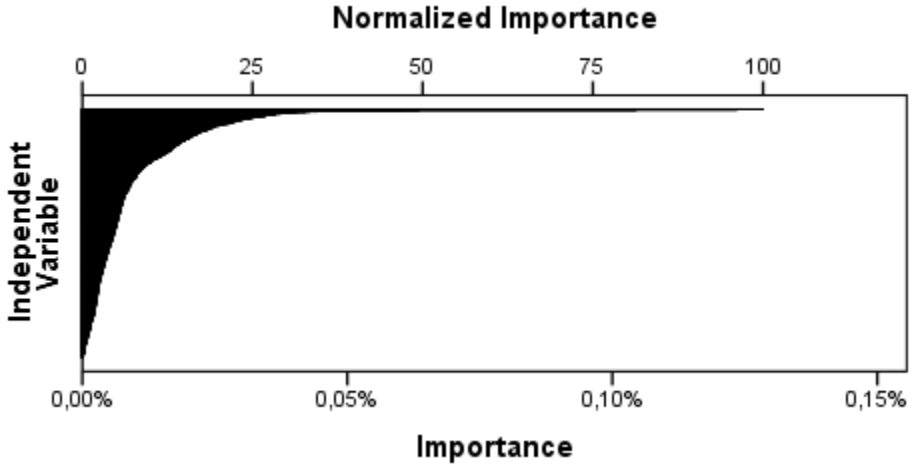

Fig. 7: Importance and Normalized Importance of Independent Variable.

5. Conclusion

Whereas the tendency is to completely automatize manufacturing processes, in some cases - e.g. for small lot-sizes or a high number of product variants - the manual work is necessary. The estimating of the cycle time is essential to the production process optimization. In this paper we showed, that our proposed method applying the skeleton and depth map data of the mass-produced and cheap Kinect sensors is appropriate to recognize human motion in an industrial environment. This was the first step for calculating theoretical life cycle time for utilization of the producing process optimization.

Acknowledgements

We thank Máté Gergó, László Kocsis, Tamás Czakó, András Takács and Attila Jáger for their assistance in the implementation and the testing.

References

[1] O. Karhu, R. Härkönen, P. Sorvali and P. Vepsäläinen, “Observing working postures in industry: Examples of OWAS application,” *Applied Ergonomics*, vol. 12, no. 1, pp. 13-17, 1981.

[2] S. Lin, et al., “Real-Time 3D Hand Gesture Recognition from Depth Image,” *Advanced Materials Research*, vol. 765, 2013.

[3] Y. Li, "Hand gesture recognition using Kinect," in *Proceedings of the IEEE 3rd International Conference on Software Engineering and Service Science (ICSESS)*, IEEE, 2012.
[4] H. Zhang, W. X. Du and H. Li, "Kinect gesture recognition for interactive system," Stanford University term paper for CS 299, 2012
[5] G. Cicirelli, et al., "A Kinect-based Gesture Recognition Approach for a Natural Human Robot Interface," *Int J Adv Robot Syst*, vol. 12, no. 22, 2015.
[6] K. K. Biswas and S. K. Basu, "Gesture recognition using Microsoft Kinect®," in *Proceedings of the 5th International Conference on Automation, Robotics and Applications (ICARA)*, IEEE, 2011.
[7] R. Mangera, "Static gesture recognition using features extracted from skeletal data," 2013.
[8] T. D. Nguyen, M. Kleinsorge and J. Krüger, "A System for Automated Live Ergonomics Assessment and Its Applications in Manufacturing," *Advances in Physical Ergonomics and Human Factors*, vol. 14, no. 1, pp. 211, 2014.
[9] M. J. Malinowski and E. Matsinos, "Comparative study of the two versions of the Microsoft Kinect® sensor in regard to the analysis of human motion," arXiv preprint arXiv:1504.00221, 2015.
[10] L. Breiman, J. H. Friedman, R. Olshen and C. J. Stone, "Classification and Regression Trees," 1984.
[11] W.-Y. Loh, "Classification and regression trees," in *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, no.1, 2011, pp. 14-23.
[12] M. Van Diepen and P. H. Franses, "Evaluating chi-squared automatic interaction detection," *Information Systems*, vol. 31, no.8, pp. 814-831, 2006.
[13] W.-Y. Loh and Y.-S. Shih, "Split selection methods for classification trees," *Statistica sinica*, pp. 815-840, 1997.
[14] P. A. Lachenbruch, "Discriminant Analysis," Encyclopedia of Statistical Sciences.