Team name	**The Ismar Group** Team 30 Track 3
Team leader name	Necati Cihan Camgoz
Team leader address, phone number and email	Perceptual Intelligence Lab. Oda:27 ETA-A Bogazici University, Bebek, Istanbul /TURKEY Phone: +90 212 359 7183
Rest of team members	Ahmet Alp Kindiroglu Prof. Lale Akarun (Advisor)
Team website URL (if any)	http://www.cmpe.boun.edu.tr/pilab/doku.php

Title of the contribution	Gesture Recognition Using Sequential Random Forests
General method description	We extract meaningful features from normalized skeleton data. We represent a single frame using a kernelized feature, containing features of that frame and neighbouring m frames. Using framewise random forest we first decide if a frame is gesture or nongesture. We then decide which gesture it belongs to. Then by fusing the results from these two classifiers, we make a framewise desicion. The results of classification can be visualized from the following link:
References	Cem Keskin, Furkan Kiraç, Yunus Emre Kara, Lale Akarun: **Hand Pose Estimation and Hand Shape Classification Using Multi-layered Randomized Decision Forests.** ECCV (6) 2012: 852-863 Marek H., Pavel C., Dikici E., Kındıroglu A. A., et al. (2010). Automatic Fingersign-to-speechtranslation system. Journal on Multimodal User Interfaces. Volume 4, Number 2, 61-79, DOI:10.1007/s12193-011-0059-3, (2011).

Describe data preprocessing techniques applied (if any)	Normalize skeleton according to hip and hip-shoulder length
Describe features used or data representation model (if any)	Skeleton coordinates, normalized skeleton coordinates, angles, relative hand position, hand movement
Data modalities used, i.e. depth, rgb, skeleton… (if any)	Skeleton only
Fusion strategy applied (if any)	Perform 21 class classification(21C) including nongestures. Perform 2 class(2C) classification for gesture nongesture. Apply median filter to both. Set 21C==gesture and 2C== nongesture frames as non-gesture. Set 2C=Gesture and 21C== nongesture as most occuring non-zero class in parameter sized window.
Dimensionality reduction technique applied (if any)	None

Temporal clustering approach (if any)	none
Temporal segmentation approach (if any)	Pad m neighbouring frames features together with current frame to add temporal information to random forest classifier
Gesture representation approach (if any)	none
Classifier used (if any)	Random forests
Large scale strategy (if any)	none

Transfer learning strategy (if any)	none
Temporal coherence and/or tracking approach considered (if any)	none
Other technique/strategy used not included in previous items (if any)	none
Method complexity analysis	O(n logn)

Qualitative advantages of the proposed solution	While lack of a methodology to adapt to temporal changes in gestures is a definite drawback, once trained with sufficiently large and diverse data, random forest outperforms seems to outperform graphical models.
Results of the comparison to other approaches (if any)	We used HMM's using the same features. We used kmeans to cluster and discretize our features. Using cluster labels with a discrete hmm and using T=35,40,and 45, we traversed the entire validation set. We couldn't break 15 percent accuracy.
Novelty degree of the solution and if is has been previously published	Similar classification approaches have been attempted using different classifiers. All of the individual algorithms we used are common and widely known approaches. However, to our knowledge no study exists using the exact same experimental setup and methodology employed in our system.

Language and implementation details (including platform, memory, parallelization requirements)	**The entire approach was implemented using matlab r2014a on Windows and Mac machines. Due to enormous memory requirements of the trained data paralelization which required duplicating the dataset was not employed in our classification method.**
Human effort required for implementation, training and validation?	2 people worked on the method for about 2 weeks. However, we spent about 2.5 months on alternative classification methods until we thought of this method.
Training/testing expended time?	Training models for training set 5-6 hours. Training models for training+validation set ~48 hours on a machine with 32Gb dedicated memory. (Did not use this model as it didnt offer significant improvement.) Testing on entire test set 1.30-2 hours including i/o.
General comments and impressions of the challenge	We loved the challenge. You have done a fine job of improving the quality of your dataset. We are currently working on Sign language recognition and we gained valuable experience by taking a short break and working on this challenge. Keep up the good work ☺