3D Gesture Recognition with Accelerometers & Gyroscopes

Salman Cheema
March 31 2014

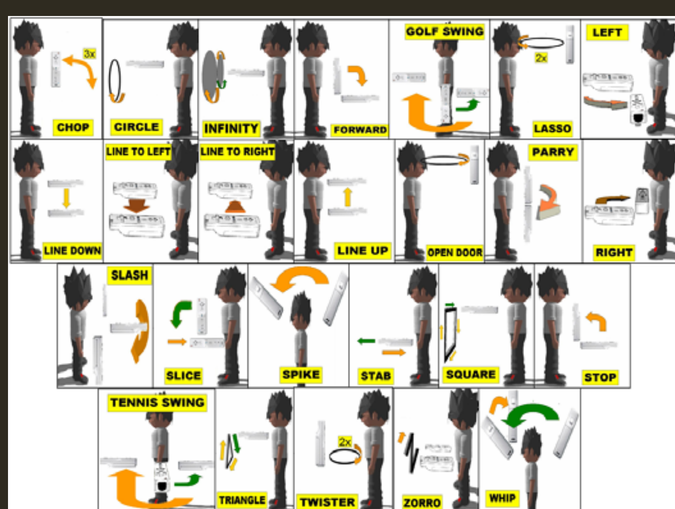
Outline

- A story of 2 final projects in Joe's 3DUI class
  - Michael Hoffman (2009)
  - Salman Cheema (2010)
- Experiments on 3d gesture recognition accuracy
  - Data collection methods
  - Analysis techniques
  - Results
  - Open Questions
- What is a 3d gesture?
  - *A motion traced in the air and detected by the use of some sensing device*

Back in 2009

- The Nintendo wiimote was supreme
  - Coolest 3d input peripheriral
  - MotionPlus had recently launched
- Acceleration and angular velocity data available
- Research Questions
  - Best classifier to use?
  - Amount of Training Data?
  - User Dependent vs User Independent?
  - Impact of additional data, i.e., angular velocity from the MotionPlus

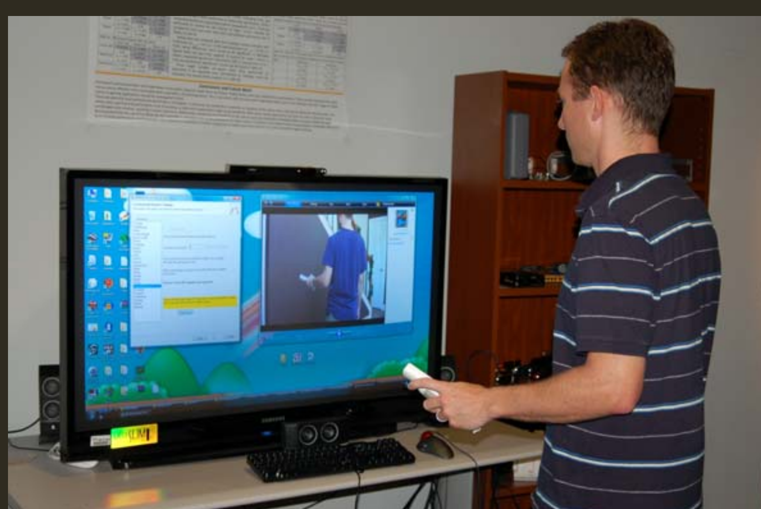
25 Gestures for Analysis



Initial Experiment

- Constructed a dataset of 8500 samples
  - 25 distinct gestures
  - 20 examples recorded per gesture
  - Collected from 17 users
  - Users shown videos to demonstrate gestures
- 2 classifiers examined
  - Linear Classifier based on Rubine's algorithm
  - AdaBoost (weak learner based on distance metric)
- Tested user-dependent & -independent training configurations

Experiment Setup



Linear Classifier

- For each distinct gesture g ϵ Set of supported gestures
  - construct a linear evaluation function
    - $F(g) = w_0 + w_1f_1 + w_2f_2 + \ldots + w_nf_n$
    - w_i is the i^{th} coefficient (weight)
    - f_i is the i^{th} feature
    - k = constant
- During training, learn weights for all evaluation functions
- Given an unknown sample
  - Compute all evaluation functions
  - The function which yields the highest value corresponds to the correct classification
- Basically, a hyper-plane in n-dimensions (n=number of features)

AdaBoost

- Use a set of weak classifiers to yield strong classification
- For training
  - Call weak learners repeatedly in a series of rounds
  - Generate a sequence of weak hypotheses
  - In each round, update weight associated with each hypothesis based on performance on training set
- Linear combination of learned weights and weak hypotheses yield a strong hypothesis

Features for Acceleration Data

- Total duration of gesture in milliseconds
- Min X, Y, Z
- Max X, Y, Z
- Median X, Y, Z
- Mean X, Y, Z
- Sine and Cosine of starting angle in XY-plane
- Sine of starting angle in XZ-plane
- Sine and Cosine of angle from first to last points in XY-plane
- Sine of angle from first to last points in XZ-plane
- Total angle traversed in XY and XZ planes
- Absolute value of total angle traversed in XY and XZ planes
- Squared value of total angle traversed in XY and XZ planes
- Length of the diagonal of the bounding volume
- Euclidean distance between first and last points
- Total distance travelled by the gesture
- Maximum acceleration squared

Features for Angular Velocity

- Min X, Y, Z
- Max X, Y, Z
- Median X, Y, Z
- Mean X, Y, Z

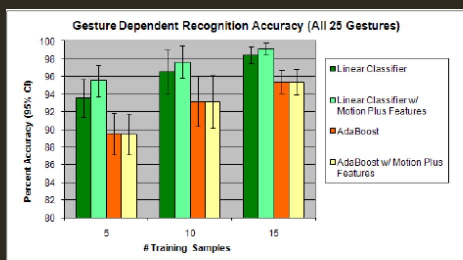
- <u>NOTE</u>: Position (x, y, z) was NOT computed from acceleration & angular velocity
  - Numerical Integration Required
  - <u>Issues</u>: Noise, Calibration, Drift

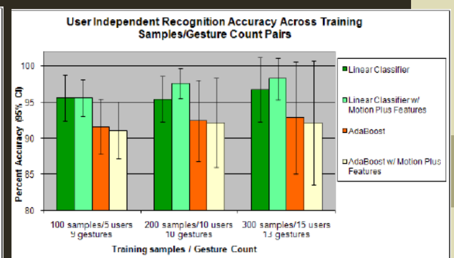
- Treated acceleration and angular velocity as position (x, y, z)
  - Works pretty well ☺

- Reduced feature set for angular velocity data
  - Singular matrices with linear classifier!!

Findings

- Achieved excellent recognition accuracy with Linear Classifier
  - >99% when trained in user-dependent mode for all gestures (trained with 15 examples/gesture)
  - >98% when trained in user-independent mode for 13 gestures (trained with 300 examples/gesture)





Criticisms

- Results sound good
- But How do they translate to a real-world setting, e.g. a videogame?

Second Round of Experiments

- Verify results of first experiment
- Examine larger set of classifiers
  - 5 classifiers tested
- Try to replicate results in different application settings
  - Data Gathering vs Video game
- Player perception of recognition accuracy
  - Important aspect of player experience

New Data Collection Experiment

- 25 people recruited from UCF
  - Collected 25 examples/gesture
  - Same set of 25 gestures
- Built an on-rails spell casting game
  - Built using Unity3D
  - Used best result from initial experiment as in-game recognition engine
  - 2 gameplay sessions per user
  - 5 maximum attempts allowed per in-game gesture
  - All attempts recorded
- Measured player perception of recognition via questionnaires

New Experiment Setup

New Dataset Constructed

- 17,890 samples
  - Contain training and gameplay examples
- 3 classes of gesture data
  - <u>Training Gestures</u>: Collected before gameplay (15,625)
  - <u>Valid Gestures</u>: Gestures correctly classified in-game (1432)
  - <u>Misclassified Gestures</u>: Gestures incorrectly classified in-game (833)

Analysis Methods

- Analyzed mean recognition accuracy in-game
- Analyzed user's questionnaire responses
- Analyzed new gesture dataset with different classifiers

In-Game Findings

- Mean accuracy significantly lower in-game
  - 69.33% in first gameplay session
  - 79% in second gameplay session
- Factors impacting lower accuracy
  - Much larger variation in gestures performed in-game
  - Players focused on gameplay
  - Unable to Recall gesture
  - Gesture Confusion
  - Possibly stressful situation

Analysis of Player Perception

- With repeated play
  - Recall improved
  - Fewer attempts needed for correct recognition
  - Players rated own performance higher
    - Possibly due to fewer attempts
  - Recognition accuracy improved (79% from 69%)
    - Players were unable to tell the difference
- Impact of low accuracy on player experience
  - Increased frustration
  - Decreased immersion
  - Distraction

Recognition Accuracy Analysis

- Tested 5 Classifiers
  - J48 Decision Trees (WEKA's implementation of the C4.5 algorithm)
  - Bayesian Networks
  - Support Vector Machines
  - Linear Classifier
  - AdaBoost (with J48 Decision Trees as weak learner)
- WEKA Toolkit used for classifier experiments
  - Except linear classifier
  - Parameters for all classifiers left at default values in WEKA

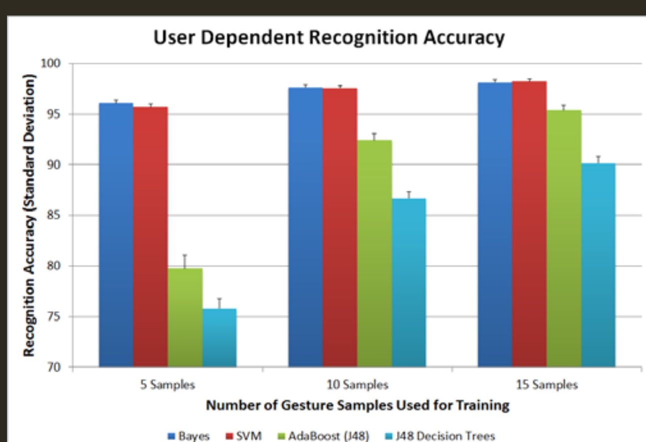
Testing Notes

- Tested both datasets
  - Michael Hoffman (2010*)
  - Salman Cheema (2013*)

*Year of publication

- Experiments in Hoffman & LaViola (2010) were highly deterministic
- For new tests
  - training examples randomly selected from available pool
  - Each experiment configuration treated as a Monte-Carlo simulation and tested 500 times
- Misclassified gestures were pruned to remove instances of failure (> 5 tries)

2010 Dataset (User Dependent)



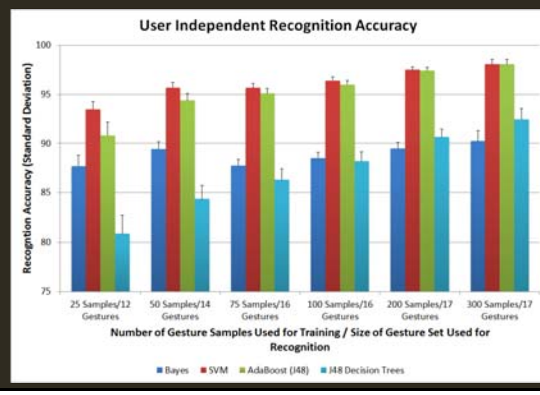
Results of Initial Experiment Confirmed: None of the new classifiers is close to 99% recognition accuracy in user-dependent configurations

2010 Dataset (User Independent)

- SVM & AdaBoost best performers
  - Larger set of gestures recognized well (98% accuracy) with less training data
  - Randomization of training data may be responsible for improvement

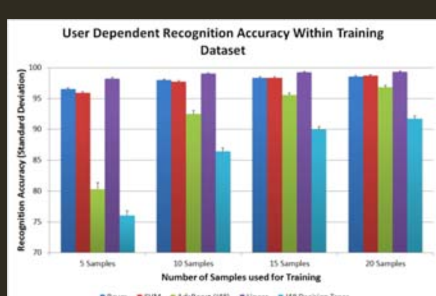
Unable to retest 2010 dataset with my linear classifier

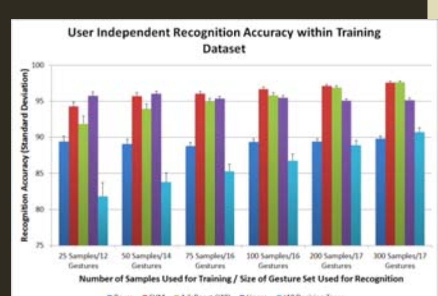
Singular Matrices



2013 Dataset : Training Gestures

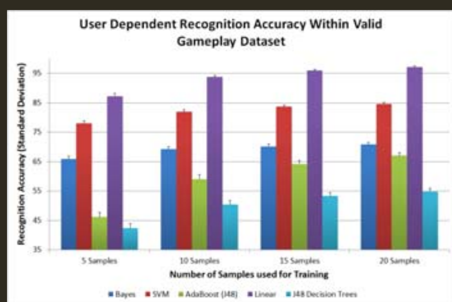
- User-Dependent: Linear Classifier still best but SVM & Bayes close
- User-Independent:
  - With less training data, Linear Classifier is best
  - With more training data, AdaBoost & SVM are best
- User-Dependent results consistent with Hoffman & LaViola.

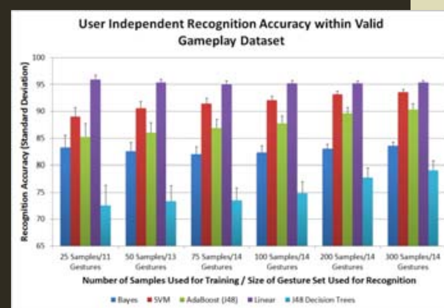




2013 Dataset : Valid Gestures

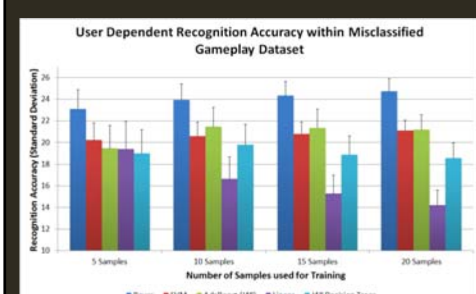
- <u>User-Dependent</u>: Linear Classifier best (97.1%)
- <u>User-Independent</u>: Linear Classifier best (95.3%)
- Interestingly, Linear Classifier is not 100% accurate
  - Greater variation in gameplay gestures
  - All training data is not created equal

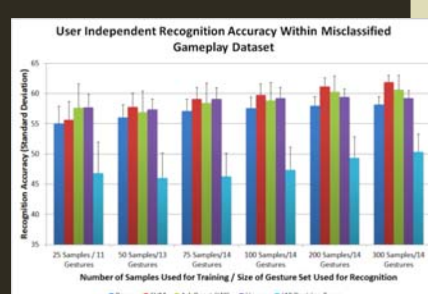




2013 Dataset : Misclassified Gesture

- <u>User-Dependent</u>: Bayes Classifier can cut error rate by one fourth
- <u>User-Independent</u>: SVM and AdaBoost can cut error rate by half
    - Data from more users may improve recognition accuracy!





Summarized Findings

- All training data is not equal
  - Fewer, higher-quality training examples have potential
  - Randomly selecting training data yielded more resilient classifiers
- Classification performance with training dataset is an upper bound
  - In-game gestures have significantly more variation
  - Application setting has a huge impact on gesture recall & confusion
- Linear Classifier came closest to upper bound in user-dependent configuration (97% vs 99%)
- Sometimes, a multi-step approach may be a good idea

Setting/Training mode	User dependent	User independent
Data collection setting	Linear	SVM or AdaBoost
Gameplay setting	Linear + Bayes	Linear + SVM

Summarized Findings (cont'd)

- Players should have been able to notice difference in recognition accuracy
  - It increased by ~10%!
- So why didn't they?
  - Focused on in-game objective
  - In-game stress
- Implication: Recognition need not be perfect?
  - A good first impression is perhaps more important
  - Impact of recognition accuracy on player experience is unclear

Research Questions

- Impact of 3D gestures on player experience still not well-examined
  - Best way to map gestures to tasks
  - Effect on experience, challenge, satisfaction, performance, immersion, ….
  - More important given new generation of gestural input systems e.g. Kinect
- How to identify best subset of training data
  - Machine learning problem
- Gesture Detection vs Gesture Recognition
  - Especially important with motion sensing systems like the Kinect

References

- Salman Cheema, Michael Hoffman, and Joseph J. LaViola Jr. 3d gesture classification with linear acceleration and angular velocity sensing devices for video games. Entertainment Computing, 4(1):11 – 24, 2013
- Salman Cheema and Joseph J. LaViola. Wizard of wii: Toward understanding player experience in first person games with 3d gestures. In Proceedings of the 6th International Conference on Foundations of Digital Games, FDG '11, pages 265–267, New York, NY, USA, 2011. ACM
- Hoffman, M., Varcholik, P., and LaViola, J. "Breaking the Status Quo: Improving 3D Gesture Recognition with Spatially Convenient Input Devices", Proceedings of IEEE Virtual Reality 2010, 59-66, March 2010
- Dean Rubine. 1991. Specifying gestures by example. *SIGGRAPH Comput. Graph.* 25, 4 (July 1991), 329-337. DOI=10.1145/127719.122753 http://doi.acm.org/10.1145/127719.122753

Q & A