GlassGesture: Exploring Head Gesture Interface of Smart Glasses
Shanhe Yi, Zhengrui Qin, Ed Novak, Yafeng Yin, and Qun Li
College of William and Mary

Introduction
Google Glass users use touchpad gestures and voice commands to interact with the device to fulfil tasks like taking picture, selecting menus, navigating the content on screen display. However, this user interface on Google Glass is less than ideal:
• Gestures on the touchpad is error-prone
• Voice cannot be applied in every scenario.
• Authentication on the touchpad is hard to use.
In GlassGesture, we uses simple gestures (Fig. 3) to control the device. The user can enter alphabets and digits by “drawing” it using the Glass. We also use those gestures as “password” to authenticate users by extracting unique features from those movements.

System Design
Efficient Similarity Search: We use DTW to match templates to recognize gestures. DTW is pair-wise which is not efficient when there is too many templates. To reduce the complexity, we build a k-d tree for KNN search.

Algorithm 1 Build KD-Tree

1. procedure BUILD_KDTREE(T, n, k)
2. for each template t in T do
3. down sampling to length-k
4. store in TDown
5. end for
6. Build a KD Tree from TDown, using Euclidean distance, as Tr
7. end procedure

Algorithm 2 KNN search.

1. procedure KNN_SEARCH(Tr, t, k)
2. put k nearest neighbors of target t in Tr into C
3. for each candidate in C do
4. run DTW on target and candidate.
5. end for
6. return index of minimal DTW distances
7. end procedure

Head Gesture based Authentication:
A “yes-or-no” question on the near-eye display will be answered using head gestures. Selected features will be extracted and put into the trained classifier.

This scheme has several advantages:
• no need to remember anything.
• taking minimal effort from user.
• more secure, hard to mimic by human eye observing.

Classification:
• Use an ensemble one-class SVM
• Grid search with constraint on the false positive rate to find a set of parameters
• Majority voting on the classification decisions from multiple SVM instances.

Evaluation
We have evaluated the system in terms of gesture recognition in different activities, gesture recognition performance and authentication performance. We have collected 6000 samples from 18 users for head gesture authentication evaluation.

(a) sitting, tiny  (b) sitting, normal  (c) walking, normal

Fig. 5 (a) Confusion matrix of command gestures (sitting, tiny). TPR: 92.87%, FPR: 5.7%. (b) Confusion matrix of command gestures (sitting, normal). TPR: 96.99%, FPR: 2.4%. (c) Confusion matrix of command gestures (walking, normal). TPR: 94.88%, FPR: 4.6%

Fig. 6 (left) Accuracy changes with sampling lengths and numbers of neighbors. Fig. 4 (mid) Accuracy and running time using DTW change with sampling length. Fig. 4 (right) running time comparison between our scheme and linear scanning. The running time will be reduced by 55% when n_DTW=10, n_Rastr=50 and k=14.

Table 1 The TPR and FPR of authentication of different gestures and comparison of other works in combining multiple instances.

<table>
<thead>
<tr>
<th>Gesture</th>
<th>TPR</th>
<th>FPR</th>
</tr>
</thead>
<tbody>
<tr>
<td>GlassGesture 1694</td>
<td>92.43% (+/-3.09)</td>
<td>0.09% (+/-0.22)</td>
</tr>
<tr>
<td>GlassGesture 1695</td>
<td>92.53% (+/-3.32)</td>
<td>0.17% (+/-0.33)</td>
</tr>
<tr>
<td>GlassGesture 1696</td>
<td>89.08% (+/-6.36)</td>
<td>0.48% (+/-0.79)</td>
</tr>
<tr>
<td>GlassGesture 1697</td>
<td>89.61% (+/-5.99)</td>
<td>0.52% (+/-0.87)</td>
</tr>
</tbody>
</table>

Table 2 Multiple and Comparison TPR and FPR

GlassGesture 1698    | 99.16% | 0.61% |
| GlassGesture 1699    | 97.14% | 1.27% |
| Touchscreen 1700    | 98.2% | 1.1% |

Fig. 7 The average TPR and FPR change with different ratios of training samples. One-class ensemble SVM (OCESM) requires less training samples with higher TPR and the cost of a small FPR loss.