

```

1: var dataCleaner = {
2:
3:   /**
4:    *
5:   */
6:   cleanVisualSearchData: function(eyeData, actionData){
7:     allData = dataCleaner.fillMissingData(eyeData);
8:     if(allData == null){
9:       return null;
10:    }
11:    eyeData = allData.eyeData;
12:
13:    var cleanData = [];
14:    var totalMissing = 0;
15:    var totalCaptured = 0;
16:    for(var i=0, k=0; i<actionData.length; i++){
17:      if(actionData[i][2]=="N"){
18:        if(actionData[i][1]=="calibrate"){
19:          i += 2;
20:        }
21:        else if(actionData[i][1]=="start"){
22:          i += 1;
23:        }
24:        var calibrate = actionData[i-2][0];
25:        var start = actionData[i-1][0];
26:        var answered = actionData[i][0];
27:        cleanData.push({calibrate:calibrate, start:start, end:answered, status:"discard",
28:          data:[]});
29:      }
30:      else{
31:        if(actionData[i][1]=="start" && actionData[i+1][2]!="N"){
32:          var group = [];
33:          var calibrate = actionData[i-1][0];
34:          var start = actionData[i][0];
35:          var answered = actionData[i+1][0];
36:          var missing = 0;
37:
38:          for(; k<eyeData.length; k++){
39:            if(Time.diff(eyeData[k][0], start) <= 0){
40:              if(Time.diff(eyeData[k][0], answered) >= 0){
41:
42:                // push one data point before the beginning of the experiment, i.e. in
43:                // calibrate section
44:                if(group.length == 0){
45:                  group.push({
46:                    time: eyeData[k-1][0],
47:                    x: eyeData[k-1][1],
48:                    y: eyeData[k-1][2],
49:                    type: "Calibrate",
50:                    from: eyeData[k][4]
51:                  });
52:                }
53:                // push the current element
54:                group.push({
55:                  time: eyeData[k][0],
56:                  x: eyeData[k][1],
57:                  y: eyeData[k][2],
58:                  type: eyeData[k][3],
59:                  from: eyeData[k][4]
60:                });
61:
62:                if(eyeData[k][4] == "missing"){
63:                  missing++;
64:                }
65:              }
66:            else{
67:              break;
68:            }
69:          }

```

```

70:         } // end for
71:
72:         cleanData.push({calibrate:calibrate, start:start, end:answered, status:"normal",
73:             data:group, numMissingPoints:missing});
74:         totalMissing += missing;
75:         totalCaptured += (group.length - missing);
76:         i++;
77:     }
78: }
79: return {trial:cleanData, numMissingPoints:totalMissing,
80:     numCapturedPoints:totalCaptured};
81: },
82:
83: /**
84:  After cleaning, the data return is in JSON:
85:  {
86:    numOfMissingPoints: (INT),
87:    numOfCapturedPoints: (INT),
88:    trial: [
89:      {
90:        start: (TIMESTAMP),
91:        end: (TIMESTAMP),
92:        data: [
93:          {
94:            time: (TIMESTAMP),
95:            type: ("SmoothPursuit"),
96:            x: (INT),
97:            y: (INT)
98:          }
99:        ],
100:      },
101:      .
102:      .
103:      .
104:    ]
105:  }
106: */
107: cleanSmoothPursuitData: function(eyeData, actionData, discardTime){
108:     allData = dataCleaner.fillMissingData(eyeData);
109:     if(allData == null){
110:         return null;
111:     }
112:     eyeData = allData.eyeData;
113:
114:     if(eyeData == null){
115:         return null;
116:     }
117:
118:     var cleanData = [];
119:     var totalMissing = 0;
120:     var totalCaptured = 0;
121:     for(var i=0, k=0; i<=40; i++){
122:         if(i == 20){
123:             i++;
124:         }
125:         var offset = i==0 || i==21 ? -100 : 0;
126:
127:         var group = [];
128:         var start = actionData[i][0];
129:         var ended = actionData[i+1][0];
130:         var movement = i < 20 ? "horizontal" : "vertical";
131:         var missing = 0;
132:
133:         for(; k<eyeData.length; k++){
134:             if(Time.diff(eyeData[k][0], start) <= offset){
135:                 if(Time.diff(eyeData[k][0], ended) > 0){
136:
137:                     // push one data point before the beginning of the experiment, i.e. in calibrate
138:                     section
139:                     if(group.length== 0){

```

```

139:         // console.log(i + " " + movement + " " + offset + " " + start + " " +
140:         eyeData[k][0]);
141:     group.push({
142:         time: eyeData[k-1][0],
143:         x: eyeData[k-1][1],
144:         y: eyeData[k-1][2],
145:         type: "Calibrate",
146:         from: eyeData[k][4]
147:     });
148: }
149:
150: // push the current element
151: group.push({
152:     time: eyeData[k][0],
153:     x: eyeData[k][1],
154:     y: eyeData[k][2],
155:     type: eyeData[k][3],
156:     from: eyeData[k][4]
157: });
158:
159: if(eyeData[k][4] == "missing"){
160:     missing++;
161: }
162: else{
163:     break;
164: }
165: }
166: }
167: cleanData.push({start:start, end:ended, data:group, movement:movement,
168:     numMissingPoints:missing});
169: totalMissing += missing;
170: totalCaptured += (group.length - missing);
171: }
172: return {numMissingPoints:totalMissing, numCapturedPoints:totalCaptured,
173:     trial:cleanData};
174: // return {numMissingPoints:allData.numMissingPoints,
175:     numCapturedPoints:allData.numCapturedPoints, trial:cleanData};
176: },
177: /**
178: Input "eye data file" and "action data file" and output "cleaned eye data".
179: The "eye data file" is a CSV. 1st column is time, 2nd column is x position, 3rd column is
180: y position.
181: LocalTimeStamp,GazePointX (ADCSpX),GazePointY (ADCSpX), ...
182: 12:21:30:601, 805, 572, ...
183: 12:21:30:605, 857, 573, ...
184: 12:21:30:611, 901, 574, ...
185: :
186: */
187: fillMissingData: function(eyeData){
188:     var fieldID = {timestamp:null, x:null, y:null, type:null};
189:     for(var k=0; k<eyeData[0].length; k++){
190:         if(eyeData[0][k] == "LocalTimeStamp"){
191:             fieldID.timestamp = k;
192:         }
193:         else if(eyeData[0][k] == "GazePointX (ADCSpX)"){
194:             fieldID.x = k;
195:         }
196:         else if(eyeData[0][k] == "GazePointY (ADCSpX)"){
197:             fieldID.y = k;
198:         }
199:         else if(eyeData[0][k] == "GazeEventType"){
200:             fieldID.type = k;
201:         }
202:     }
203: }
204: if(fieldID.timestamp==null || fieldID.x==null || fieldID.y==null || fieldID.type==null){
205:     return null;

```

```

205:     }
206:
207:     var numCaptured = 0;
208:     var numMissing = 0;
209:
210:     // fill the missing data
211:     var eyeDataClean = [];
212:     for(var i=1; i<eyeData.length; i++){
213:         if(eyeData[i][fieldID.x]!="" && eyeData[i][fieldID.y]!=""){
214:             numCaptured++;
215:             eyeDataClean.push([eyeData[i][fieldID.timestamp], parseInt(eyeData[i][fieldID.x]),
216:                 parseInt(eyeData[i][fieldID.y]), eyeData[i][fieldID.type], "recorded"]);
217:         }
218:         else{
219:             if(eyeDataClean.length >= 1){
220:                 var missingBegin = i-1;
221:                 var missingEnd = null;
222:                 for(; i<eyeData.length; i++){
223:                     if(eyeData[i][fieldID.x]!="" && eyeData[i][fieldID.y]!=" && typeof(eyeData[i]
224:                         [fieldID.x])!="undefined" && typeof(eyeData[i][fieldID.y])!="undefined"){
225:                         missingEnd = i;
226:                         break;
227:                     }
228:                 }
229:                 if(missingEnd != null){
230:                     var missingCount = missingEnd - missingBegin - 1;
231:                     numMissing += missingCount;
232:
233:                     var t1 = Time.parseInt(eyeData[missingBegin][fieldID.timestamp]);
234:                     var x1 = parseInt(eyeData[missingBegin][fieldID.x]);
235:                     var y1 = parseInt(eyeData[missingBegin][fieldID.y]);
236:
237:                     var t2 = Time.parseInt(eyeData[missingEnd][fieldID.timestamp]);
238:                     var x2 = parseInt(eyeData[missingEnd][fieldID.x]);
239:                     var y2 = parseInt(eyeData[missingEnd][fieldID.y]);
240:
241:                     var tDelta = (t2-t1)/(missingCount + 1);
242:                     var xDelta = (x2-x1)/(missingCount + 1);
243:                     var yDelta = (y2-y1)/(missingCount + 1);
244:
245:                     for(var j=1; j<=missingCount; j++){
246:                         var missingData = eyeData[missingBegin + j];
247:
248:                         var t = Math.round(t1 + tDelta * j);
249:                         var x = Math.round(x1 + xDelta * j);
250:                         var y = Math.round(y1 + yDelta * j);
251:
252:                         var time = Time.parseStr(t);
253:
254:                         eyeDataClean.push([time, x, y, eyeData[missingBegin + j][fieldID.type],
255:                             "missing"]);
256:                     }
257:                     eyeDataClean.push([eyeData[i][fieldID.timestamp], parseInt(eyeData[i]
258:                         [fieldID.x]), parseInt(eyeData[i][fieldID.y]), eyeData[i][fieldID.type],
259:                             "missing"]);
260:                 }
261:             }
262:         }
263:     }
264:     return {eyeData:eyeDataClean, numMissingPoints:numMissing,
265:         numCapturedPoints:numCaptured};
266: },
267:
268: /**
269:  *
270:  */
271: printEyeData: function(data, element){
272:     var str = "";
273:     for(var i=0; i<data.length; i++){
274:         str += data[i][0]+"\\t"+data[i][1]+"\\t"+data[i][2]+"\\t"+data[i][3]+"\\n";

```

```
270:     }
271:     if(typeof(element) != "undefined"){
272:         $("#tmp").val(str);
273:     }
274:     else{
275:         console.log(str);
276:     }
277: }
278:
279: }
280:
```