

```

1  /*****
2  File Name:      FIXFUNC.C
3  Program Name:   Eye Fixation Analysis Functions
4  Company:        LC Technologies, Inc.
5                  9455 Silver King Court
6                  Fairfax, VA 22031
7                  (703) 385-7133
8  Date Created:   10/20/89
9                  04/12/95 modified: turned into collection of functions
10 *****/
11
12 /*****
13 /* FUNCTION, VARIABLE AND CONSTANT DEFINITIONS: */
14
15 #include <stdio.h>
16 #include <conio.h>
17 #include <math.h>
18 #include <string.h>
19 #include <process.h>
20 #include "lctypdef.h"
21 // #include <lctsupt.h>
22 #include "fixfunc.h"
23 // #include <crtdbg.h>          /* debug tools */
24
25 #define RING_SIZE      1251    /* length of the delay line in */
26                                /* DetectFixation() -- */
27                                /* should be greater than */
28                                /* iMinimumFixSamples */
29
30 #define NEW_FIX      0
31 #define PRES_FIX     1
32 #define PREV_FIX     2
33
34 /*****
35 /* FUNCTION PROTOTYPES: */
36
37 void ResetFixation(int iNewPresPrev);
38 void StartFixAtGazepoint(int iNewPresPrev, float fXGaze, float fYGaze);
39 void UpdateFixation(int iNewPresPrev,
40                    float fXGaze, float fYGaze, int iMinimumFixSamples);
41 float CalcGazeDeviationFromFix(int iNewPresPrev, float fXGaze, float fYGaze);
42 void CheckIfFixating(int iMinimumFixSamples);
43 void MoveNewFixToPresFix(int iMinimumFixSamples);
44 void DeclareCompletedFixation(int iMinimumFixSamples, int iSamplesAgo);
45 void RestoreOutPoints(void);
46
47 /*****
48 /* STRUCTURES DEFINITIONS: */
49
50 struct _stFix          /* FIXATION DATA */
51 {
52     int    iStartCount; /* call count that the fixation starts */
53     int    iEndCount;   /* call count that the fixation ends */
54     int    iNSamples;   /* number of samples in the present fix */
55     float  fXSum;        /* summations for calculation of average */
56     float  fYSum;        /* fixation position */
57     float  fX;           /* average coordinate of the eye fixation */
58     float  fY;           /* point (user selected units) */
59 };
60
61 struct _stRingBuf      /* RING BUFFER STORING PAST VALUES: */
62 {
63     float  fXGaze;       /* gazepoint coordinate */
64     float  fYGaze;       /*
65     BOOL    bGazeFound;  /* gazepoint found flag
66     int     iEyeMotionState; /* state of the eye motion:
67                                /* MOVING
68                                /* FIXATING
69                                /* FIXATION_COMPLETED
70     float  fXFix;        /* current fixation coordinate
71     float  fYFix;        /*
72     float  fGazeDeviation; /* gazepoint deviation from the fixation

```

```

72     int    iSacDuration;          /* saccade duration          */
73     int    iFixDuration;          /* fixation duration        */
74 };
75
76 /*****
77 /* GLOBAL FIXFUNC VARIABLES:
78
79 int    iCallCount;                /* number of times this function has been
80                                     /* called since it was initialized
81                                     /* (30ths or 60ths of a second,
82                                     /* depending on eyetracking sample rate)
83 int    iNPresOut;                /* number of successive gaze point samples
84                                     /* outside the present fixation circle
85
86 int    iNSamplesSinceLastGoodFixPoint;
87                                     /* number of samples since the last known
88                                     /* good gaze point in the present fixation
89                                     /* Note: During continuous tracking of a
90                                     /* normal fixation, this value is 1.
91 float  fPresDr;                  /* difference between gaze point and
92                                     /* fixation (x, y, and radius)
93 float  fNewDr;                  /* difference between gaze point and
94                                     /* fixation (x, y, and radius)
95 int    iMaxMissedSamples;        /* maximum number of successive gaze point
96                                     /* samples that may go untracked before
97                                     /* a fixation is allowed to continue
98 int    iMaxOutSamples;          /* maximum number of successive gaze point
99                                     /* samples that may go outside the
100                                     /* fixation circle
101 struct _stFix stFix[3];         /* prior, present and new fixations
102                                     /* indexed by: NEW_FIX 0
103                                     /* PRES_FIX 1
104                                     /* PREV_FIX 2
105 struct _stRingBuf stRingBuf[RING_SIZE];
106 int    iRingIndex;              /* ring index of the present gaze sample
107 int    iRingIndexDelay;         /* ring index of the gaze sample taken
108                                     /* iMinimumFixSamples ago
109
110 #if defined DIAG_FIXFUNC
111     FILE *hfFixFile;
112 #endif /* DIAG_FIXFUNC */
113
114 /*****
115 void InitFixation(int iMinimumFixSamples)
116                                     /* minimum number of gaze samples
117                                     /* that can be considered a
118                                     /* fixation
119                                     /* Note: if the input value is
120                                     /* is less than 3, the function
121                                     /* sets it to 3
122
123 /* This function clears any previous, present and new fixations, and it
124 /* initializes DetectFixation()'s internal ring buffers of prior
125 /* gaze point data. InitFixation() should be called prior to a sequence
126 /* of calls to DetectFixation().
127
128 {
129 /* Set the maximum allowable number of consecutive samples that may go
130 /* untracked within a fixation.
131     iMaxMissedSamples = 3;
132
133 /* Set the maximum allowable number of consecutive samples that may go
134 /* outside the fixation circle.
135     iMaxOutSamples = 1;
136
137 /* Initialize the internal ring buffer.
138     for (iRingIndex = 0; iRingIndex < RING_SIZE; iRingIndex++)
139     {
140         stRingBuf[iRingIndex].fXGaze    = 0.0F;
141         stRingBuf[iRingIndex].fYGaze    = 0.0F;
142

```

(Just the first two pages of code are shown here.)