

Reference Manual

Generated by Doxygen 1.5.6

Wed Aug 6 02:51:14 2008

Contents

1	File Index	1
1.1	File List	1
2	File Documentation	3
2.1	ehci.cpp File Reference	3
2.1.1	Define Documentation	4
2.1.1.1	NUMPTS	4
2.1.1.2	READFROMIMAGEFILE	4
2.1.2	Function Documentation	4
2.1.2.1	detect_and_draw	4
2.1.2.2	ehciExit	5
2.1.2.3	ehciInit	5
2.1.2.4	ehciLoop	5
2.1.2.5	getCurrentFrame	5
2.1.2.6	getGLPositMatrix	5
2.1.2.7	getHeadBounds	5
2.1.2.8	getHeadPosition	5
2.1.2.9	getPositMatrix	5
2.1.2.10	getReferenceHeadBounds	5
2.1.2.11	initializeCapture	6
2.1.2.12	insertNewPoints	6
2.1.2.13	loadCascade	6
2.1.2.14	printMatrixData	6
2.1.2.15	setGLProjectionMatrix	6
2.1.2.16	setInitialRTMatrix	6
2.1.2.17	update6dof	6
2.1.2.18	updateGLPositMatrix	7
2.1.2.19	updateInternalHeadPosition	7

2.1.2.20	updateReferenceInternalHeadPosition	7
2.1.3	Variable Documentation	7
2.1.3.1	capture	7
2.1.3.2	frame	7
2.1.3.3	glPositMatrix	7
2.1.3.4	grey	7
2.1.3.5	image	7
2.1.3.6	lastHeadH	7
2.1.3.7	lastHeadW	7
2.1.3.8	MAX_COUNT	7
2.1.3.9	modelPoints	7
2.1.3.10	myHeadHeight	7
2.1.3.11	myHeadWidth	7
2.1.3.12	myUpperHeadX	7
2.1.3.13	myUpperHeadY	8
2.1.3.14	points	8
2.1.3.15	prev_grey	8
2.1.3.16	prev_pyramid	8
2.1.3.17	pyramid	8
2.1.3.18	referenceHeadHeight	8
2.1.3.19	referenceHeadWidth	8
2.1.3.20	referenceUpperHeadX	8
2.1.3.21	referenceUpperHeadY	8
2.1.3.22	status	8
2.1.3.23	swap_points	8
2.1.3.24	swap_temp	8
2.1.3.25	upperHeadCorner	8
2.1.3.26	win_size	8
2.2	ehci.dox File Reference	9
2.3	ehci.h File Reference	10
2.3.1	Define Documentation	10
2.3.1.1	EHCIFOCUS	10
2.3.1.2	EHCIMODELSCALE	10
2.3.2	Function Documentation	10
2.3.2.1	detect_and_draw	10
2.3.2.2	ehciExit	11

2.3.2.3	ehciInit	11
2.3.2.4	ehciLoop	11
2.3.2.5	getCurrentFrame	11
2.3.2.6	getGlPositMatrix	11
2.3.2.7	getHeadBounds	11
2.3.2.8	getHeadPosition	11
2.3.2.9	getPositMatrix	11
2.3.2.10	getReferenceHeadBounds	11
2.3.2.11	insertNewPoints	12
2.3.2.12	setGLProjectionMatrix	12
2.3.2.13	setInitialRTMatrix	12
2.3.2.14	updateGlPositMatrix	12
2.3.3	Variable Documentation	12
2.3.3.1	EHCI2DFACEDETECT	12
2.3.3.2	EHCI6DFACEDETECT	12

Chapter 1

File Index

1.1 File List

Here is a list of all files with brief descriptions:

ehci.cpp	3
ehci.h	10

Chapter 2

File Documentation

2.1 ehci.cpp File Reference

```
#include "ehci.h"
#include <stdio.h>
#include <vector>
```

Defines

- #define [READFROMIMAGEFILE](#) 0
- #define [NUMPTS](#) 8

Functions

- void [loadCascade](#) (CvHaarClassifierCascade **cascade)
- int [detect_and_draw](#) (IplImage *img, CvPoint *[upperHeadCorner](#), int *headWidth, int *headHeight, CvHaarClassifierCascade *cascade, CvMemStorage *storage)
- int [getHeadPosition](#) (IplImage *frame, CvPoint *[upperHeadCorner](#), int *headWidth, int *headHeight)
- void [updateGlPositMatrix](#) (CvMatr32f rotation_matrix, CvVect32f translation_vector)
- void [getGlPositMatrix](#) (double myGlPositMatrix[16])
- void [setInitialRTMatrix](#) (CvMatr32f rotation_matrix, CvVect32f translation_vector)
- void [printMatrixData](#) (CvMatr32f rotation_matrix, CvVect32f translation_vector)
- void [getPositMatrix](#) (IplImage *myImage, int initialGuess, CvMatr32f rotation_matrix, CvVect32f translation_vector, int numOfTrackingPoints, int focus, CvPoint2D32f *[points](#), CvPoint [upperHeadCorner](#), int headWidth, int headHeight, float modelScale)
- int [insertNewPoints](#) (IplImage *[grey](#), int headX, int headY, int width, int height, CvPoint2D32f *[points](#))
- void [setGLProjectionMatrix](#) (double projectionMatrix[16])
- IplImage * [getCurrentFrame](#) ()
- int [initializeCapture](#) ()
- void [update6dof](#) (int headHeight, int headWidth, int initialGuess, int numberOfTrackingPoints)
- void [updateInternalHeadPosition](#) (int upperHeadX, int upperHeadY, int headWidth, int headHeight)

- void [updateReferenceInternalHeadPosition](#) (int upperHeadX, int upperHeadY, int headWidth, int headHeight)
- void [getReferenceHeadBounds](#) (int *headRefX, int *headRefY, int *aLastHeadW, int *aLastHeadH)
- void [getHeadBounds](#) (int *headRefX, int *headRefY, int *aLastHeadW, int *aLastHeadH)
- void [ehciInit](#) ()
- int [ehciLoop](#) (int mode, int initialGuess)
- void [ehciExit](#) ()

Variables

- double [glPositMatrix](#) [16]
- std::vector< CvPoint3D32f > [modelPoints](#)
- IplImage * [image](#) = 0
- IplImage * [grey](#) = 0
- IplImage * [prev_grey](#) = 0
- IplImage * [pyramid](#) = 0
- IplImage * [prev_pyramid](#) = 0
- IplImage * [swap_temp](#)
- const int [MAX_COUNT](#) = 500
- CvPoint2D32f * [points](#) [2] = {0,0}
- CvPoint2D32f * [swap_points](#)
- char * [status](#) = 0
- CvPoint [upperHeadCorner](#) = cvPoint(0,0)
- int [lastHeadW](#)
- int [lastHeadH](#)
- int [win_size](#) = 10
- IplImage * [frame](#)
- CvCapture * [capture](#) = 0
- int [myUpperHeadX](#)
- int [myUpperHeadY](#)
- int [myHeadWidth](#)
- int [myHeadHeight](#)
- int [referenceUpperHeadX](#)
- int [referenceUpperHeadY](#)
- int [referenceHeadWidth](#)
- int [referenceHeadHeight](#)

2.1.1 Define Documentation

2.1.1.1 #define NUMPTS 8

2.1.1.2 #define READFROMIMAGEFILE 0

2.1.2 Function Documentation

2.1.2.1 int detect_and_draw (IplImage * *img*, CvPoint * *upperHeadCorner*, int * *headWidth*, int * *headHeight*, CvHaarClassifierCascade * *cascade*, CvMemStorage * *storage*)

Optimized function to detect head position and draw a rectangle around it. This function uses OpenCV implemented Viola-Jones algorithm with Haar-like features and trained cascades. Refer to OpenCV documentation.

2.1.2.2 void ehciExit ()

ehci cleanup code

2.1.2.3 void ehciInit ()

Deals with library initialization and creating debug windows.

2.1.2.4 int ehciLoop (int *mode*, int *initialGuess*)

2.1.2.5 IplImage* getCurrentFrame ()

Returns last captured frame.

2.1.2.6 void getGlPositMatrix (double *myGlPositMatrix*[16])

Copies currently detected posit matrix (translation and rotation) in the openGl format to the parameter array

2.1.2.7 void getHeadBounds (int * *headRefX*, int * *headRefY*, int * *aLastHeadW*, int * *aLastHeadH*)

Returns last captured head position in pixel dimensions. Upper left is x=0, y = 0 . Head width and height are also given in pixels.

2.1.2.8 int getHeadPosition (IplImage * *frame*, CvPoint * *upperHeadCorner*, int * *headWidth*, int * *headHeight*)

internal ehci function to get 2d head upper left corner, width and height the width and height are proportional to Viola-Jones trained cascade which are a little bit smaller than the real ones

returns 0 if no head was found

2.1.2.9 void getPositMatrix (IplImage * *myImage*, int *initialGuess*, CvMatr32f *rotation_matrix*, CvVect32f *translation_vector*, int *numOfTrackingPoints*, int *focus*, CvPoint2D32f * *points*, CvPoint *upperHeadCorner*, int *headWidth*, int *headHeight*, float *modelScale*)

This function will retrieve the rotation and translation matrixes using the POSIT algorithm In case the initialGuess parameter is set to 1, the algorithm will map points to the sinoidal head, else it will only track to the original ones. In a future version, this function should also map new points back to the current head position

2.1.2.10 void getReferenceHeadBounds (int * *headRefX*, int * *headRefY*, int * *aLastHeadW*, int * *aLastHeadH*)

Returns reference head position in pixel dimensions. Upper left is x=0, y = 0 . Head width and height are also given in pixels.

2.1.2.11 int initializeCapture ()**2.1.2.12 int insertNewPoints (IplImage * *grey*, int *headX*, int *headY*, int *width*, int *height*, CvPoint2D32f * *points*)**

This function inserts feature points according to cvFindGoodFeatures to Track in the roi given by headX,headY, width, and height Returns the number of points it was able to insert

2.1.2.13 void loadCascade (CvHaarClassifierCascade ** *cascade*)

Loads cascade xml

2.1.2.14 void printMatrixData (CvMatr32f *rotation_matrix*, CvVect32f *translation_vector*)**2.1.2.15 void setGLProjectionMatrix (double *projectionMatrix*[16])****2.1.2.16 void setInitialRTMatrix (CvMatr32f *rotation_matrix*, CvVect32f *translation_vector*)****2.1.2.17 void update6dof (int *headHeight*, int *headWidth*, int *initialGuess*, int *numberOfTrackingPoints*)**

Updates 6 degrees of freedom head tracking.

2.1.2.18 void updateGIPositMatrix (CvMatr32f *rotation_matrix*, CvVect32f *translation_vector*)

2.1.2.19 void updateInternalHeadPosition (int *upperHeadX*, int *upperHeadY*, int *headWidth*, int *headHeight*)

2.1.2.20 void updateReferenceInternalHeadPosition (int *upperHeadX*, int *upperHeadY*, int *headWidth*, int *headHeight*)

2.1.3 Variable Documentation

2.1.3.1 CvCapture* *capture* = 0

2.1.3.2 IplImage* *frame*

2.1.3.3 double *glPositMatrix*[16]

2.1.3.4 IplImage * *grey* = 0

2.1.3.5 IplImage* *image* = 0

2.1.3.6 int *lastHeadH*

2.1.3.7 int *lastHeadW*

2.1.3.8 const int *MAX_COUNT* = 500

2.1.3.9 std::vector<CvPoint3D32f> *modelPoints*

2.1.3.10 int *myHeadHeight*

2.1.3.11 int *myHeadWidth*

2.1.3.12 int *myUpperHeadX*

Updates internal parameters so that `getHeadParameters` works accordingly.

- 2.1.3.13 `int myUpperHeadY`
- 2.1.3.14 `CvPoint2D32f* points[2] = {0,0}`
- 2.1.3.15 `IplImage * prev_grey = 0`
- 2.1.3.16 `IplImage * prev_pyramid = 0`
- 2.1.3.17 `IplImage * pyramid = 0`
- 2.1.3.18 `int referenceHeadHeight`
- 2.1.3.19 `int referenceHeadWidth`
- 2.1.3.20 `int referenceUpperHeadX`
- 2.1.3.21 `int referenceUpperHeadY`
- 2.1.3.22 `char* status = 0`
- 2.1.3.23 `CvPoint2D32f * swap_points`
- 2.1.3.24 `IplImage * swap_temp`
- 2.1.3.25 `CvPoint upperHeadCorner = cvPoint(0,0)`
- 2.1.3.26 `int win_size = 10`

2.2 ehci.dox File Reference

2.3 ehci.h File Reference

```
#include "cv.h"
#include "highgui.h"
```

Defines

- `#define EHCIMODELSCALE 50`
- `#define EHCIFOCUS 602`

Functions

- void [updateGIPositMatrix](#) (CvMatr32f rotation_matrix, CvVect32f translation_vector)
- void [setInitialRTMatrix](#) (CvMatr32f rotation_matrix, CvVect32f translation_vector)
- int [getHeadPosition](#) (IplImage *frame, CvPoint *upperHeadCorner, int *headWidth, int *headHeight)
- int [detect_and_draw](#) (IplImage *img, CvPoint *upperHeadCorner, int *headWidth, int *headHeight, CvHaarClassifierCascade *cascade, CvMemStorage *storage)
- void [getPositMatrix](#) (IplImage *myImage, int initialGuess, CvMatr32f rotation_matrix, CvVect32f translation_vector, int numOfTrackingPoints, int focus, CvPoint2D32f *points, CvPoint upperHeadCorner, int headWidth, int headHeight, float modelScale)
- int [insertNewPoints](#) (IplImage *grey, int headX, int headY, int width, int height, CvPoint2D32f *points)
- void [setGLProjectionMatrix](#) (double projectionMatrix[16])
- void [ehciInit](#) ()
- int [ehciLoop](#) (int mode, int initialGuess)
- void [getHeadBounds](#) (int *headRefX, int *headRefY, int *aLastHeadW, int *aLastHeadH)
- void [getReferenceHeadBounds](#) (int *headRefX, int *headRefY, int *aLastHeadW, int *aLastHeadH)
- void [getGIPositMatrix](#) (double myGIPositMatrix[16])
- IplImage * [getCurrentFrame](#) ()
- void [ehciExit](#) ()

Variables

- int [EHCI2DFACEDETECT](#) = 1
- int [EHCI6DFACEDETECT](#) = 3

2.3.1 Define Documentation

2.3.1.1 `#define EHCIFOCUS 602`

2.3.1.2 `#define EHCIMODELSCALE 50`

2.3.2 Function Documentation

2.3.2.1 int [detect_and_draw](#) (IplImage *img, CvPoint *upperHeadCorner, int *headWidth, int *headHeight, CvHaarClassifierCascade *cascade, CvMemStorage *storage)

Optimized function to detect head position and draw a rectangle around it. This function uses OpenCV implemented Viola-Jones algorithm with Haar-like features and trained cascades. Refer to OpenCV docu-

mentation.

2.3.2.2 void ehciExit ()

ehci cleanup code

2.3.2.3 void ehciInit ()

Deals with library initialization and creating debug windows.

2.3.2.4 int ehciLoop (int mode, int initialGuess)

2.3.2.5 IplImage* getCurrentFrame ()

Returns last captured frame.

2.3.2.6 void getGlPositMatrix (double myGlPositMatrix[16])

Copies currently detected posit matrix (translation and rotation) in the openGl format to the parameter array

2.3.2.7 void getHeadBounds (int * headRefX, int * headRefY, int * aLastHeadW, int * aLastHeadH)

Returns last captured head position in pixel dimensions. Upper left is x=0, y = 0 . Head width and height are also given in pixels.

2.3.2.8 int getHeadPosition (IplImage * frame, CvPoint * upperHeadCorner, int * headWidth, int * headHeight)

internal ehci function to get 2d head upper left corner, width and height the width and height are proportional to Viola-Jones trained cascade which are a little bit smaller than the real ones

returns 0 if no head was found

2.3.2.9 void getPositMatrix (IplImage * myImage, int initialGuess, CvMatr32f rotation_matrix, CvVect32f translation_vector, int numOfTrackingPoints, int focus, CvPoint2D32f * points, CvPoint upperHeadCorner, int headWidth, int headHeight, float modelScale)

This function will retrieve the rotation and translation matrixes using the POSIT algorithm In case the initialGuess parameter is set to 1, the algorithm will map points to the sinoidal head, else it will only track to the original ones. In a future version, this function should also map new points back to the current head position

2.3.2.10 void getReferenceHeadBounds (int * headRefX, int * headRefY, int * aLastHeadW, int * aLastHeadH)

Returns reference head position in pixel dimensions. Upper left is x=0, y = 0 . Head width and height are also given in pixels.

2.3.2.11 int insertNewPoints (IplImage * *grey*, int *headX*, int *headY*, int *width*, int *height*, CvPoint2D32f * *points*)

This function inserts feature points according to cvFindGoodFeatures to Track in the roi given by headX,headY, width, and height Returns the number of points it was able to insert

2.3.2.12 void setGLProjectionMatrix (double *projectionMatrix*[16])**2.3.2.13 void setInitialRTMatrix (CvMatr32f *rotation_matrix*, CvVect32f *translation_vector*)****2.3.2.14 void updateGIPositMatrix (CvMatr32f *rotation_matrix*, CvVect32f *translation_vector*)****2.3.3 Variable Documentation****2.3.3.1 int EHCI2DFACEDETECT = 1**

ehci loop modes EHCI2DFACEDETECT - only makes 2d facedetect EHCI6DFACEDETECT - makes 2d and 6dof facedetection, so it is 0x00000001 | 0x00000010

2.3.3.2 int EHCI6DFACEDETECT = 3

Index

- capture
 - [ehci.cpp](#), [7](#)
- detect_and_draw
 - [ehci.cpp](#), [4](#)
 - [ehci.h](#), [10](#)
- [ehci.cpp](#), [3](#)
 - [capture](#), [7](#)
 - [detect_and_draw](#), [4](#)
 - [ehciExit](#), [4](#)
 - [ehciInit](#), [5](#)
 - [ehciLoop](#), [5](#)
 - [frame](#), [7](#)
 - [getCurrentFrame](#), [5](#)
 - [getGLPositMatrix](#), [5](#)
 - [getHeadBounds](#), [5](#)
 - [getHeadPosition](#), [5](#)
 - [getPositMatrix](#), [5](#)
 - [getReferenceHeadBounds](#), [5](#)
 - [glPositMatrix](#), [7](#)
 - [grey](#), [7](#)
 - [image](#), [7](#)
 - [initializeCapture](#), [5](#)
 - [insertNewPoints](#), [6](#)
 - [lastHeadH](#), [7](#)
 - [lastHeadW](#), [7](#)
 - [loadCascade](#), [6](#)
 - [MAX_COUNT](#), [7](#)
 - [modelPoints](#), [7](#)
 - [myHeadHeight](#), [7](#)
 - [myHeadWidth](#), [7](#)
 - [myUpperHeadX](#), [7](#)
 - [myUpperHeadY](#), [7](#)
 - [NUMPTS](#), [4](#)
 - [points](#), [8](#)
 - [prev_grey](#), [8](#)
 - [prev_pyramid](#), [8](#)
 - [printMatrixData](#), [6](#)
 - [pyramid](#), [8](#)
 - [READFROMIMAGEFILE](#), [4](#)
 - [referenceHeadHeight](#), [8](#)
 - [referenceHeadWidth](#), [8](#)
 - [referenceUpperHeadX](#), [8](#)
 - [referenceUpperHeadY](#), [8](#)
 - [setGLProjectionMatrix](#), [6](#)
 - [setInitialRTMatrix](#), [6](#)
 - [status](#), [8](#)
 - [swap_points](#), [8](#)
 - [swap_temp](#), [8](#)
 - [update6dof](#), [6](#)
 - [updateGLPositMatrix](#), [6](#)
 - [updateInternalHeadPosition](#), [7](#)
 - [updateReferenceInternalHeadPosition](#), [7](#)
 - [upperHeadCorner](#), [8](#)
 - [win_size](#), [8](#)
- [ehci.dox](#), [9](#)
- [ehci.h](#), [10](#)
 - [detect_and_draw](#), [10](#)
 - [EHCI2DFACEDETECT](#), [12](#)
 - [EHCI6DFACEDETECT](#), [12](#)
 - [ehciExit](#), [11](#)
 - [EHCIFOCUS](#), [10](#)
 - [ehciInit](#), [11](#)
 - [ehciLoop](#), [11](#)
 - [EHCIMODELSCALE](#), [10](#)
 - [getCurrentFrame](#), [11](#)
 - [getGLPositMatrix](#), [11](#)
 - [getHeadBounds](#), [11](#)
 - [getHeadPosition](#), [11](#)
 - [getPositMatrix](#), [11](#)
 - [getReferenceHeadBounds](#), [11](#)
 - [insertNewPoints](#), [11](#)
 - [setGLProjectionMatrix](#), [12](#)
 - [setInitialRTMatrix](#), [12](#)
 - [updateGLPositMatrix](#), [12](#)
- [EHCI2DFACEDETECT](#)
 - [ehci.h](#), [12](#)
- [EHCI6DFACEDETECT](#)
 - [ehci.h](#), [12](#)
- [ehciExit](#)
 - [ehci.cpp](#), [4](#)
 - [ehci.h](#), [11](#)
- [EHCIFOCUS](#)
 - [ehci.h](#), [10](#)
- [ehciInit](#)
 - [ehci.cpp](#), [5](#)
 - [ehci.h](#), [11](#)
- [ehciLoop](#)
 - [ehci.cpp](#), [5](#)

- ehci.h, 11
- EHCIMODELSCALE
 - ehci.h, 10
- frame
 - ehci.cpp, 7
- getCurrentFrame
 - ehci.cpp, 5
 - ehci.h, 11
- getGlPositMatrix
 - ehci.cpp, 5
 - ehci.h, 11
- getHeadBounds
 - ehci.cpp, 5
 - ehci.h, 11
- getHeadPosition
 - ehci.cpp, 5
 - ehci.h, 11
- getPositMatrix
 - ehci.cpp, 5
 - ehci.h, 11
- getReferenceHeadBounds
 - ehci.cpp, 5
 - ehci.h, 11
- glPositMatrix
 - ehci.cpp, 7
- grey
 - ehci.cpp, 7
- image
 - ehci.cpp, 7
- initializeCapture
 - ehci.cpp, 5
- insertNewPoints
 - ehci.cpp, 6
 - ehci.h, 11
- lastHeadH
 - ehci.cpp, 7
- lastHeadW
 - ehci.cpp, 7
- loadCascade
 - ehci.cpp, 6
- MAX_COUNT
 - ehci.cpp, 7
- modelPoints
 - ehci.cpp, 7
- myHeadHeight
 - ehci.cpp, 7
- myHeadWidth
 - ehci.cpp, 7
- myUpperHeadX
 - ehci.cpp, 7
- myUpperHeadY
 - ehci.cpp, 7
- NUMPTS
 - ehci.cpp, 4
- points
 - ehci.cpp, 8
- prev_grey
 - ehci.cpp, 8
- prev_pyramid
 - ehci.cpp, 8
- printMatrixData
 - ehci.cpp, 6
- pyramid
 - ehci.cpp, 8
- READFROMIMAGEFILE
 - ehci.cpp, 4
- referenceHeadHeight
 - ehci.cpp, 8
- referenceHeadWidth
 - ehci.cpp, 8
- referenceUpperHeadX
 - ehci.cpp, 8
- referenceUpperHeadY
 - ehci.cpp, 8
- setGLProjectionMatrix
 - ehci.cpp, 6
 - ehci.h, 12
- setInitialRTMatrix
 - ehci.cpp, 6
 - ehci.h, 12
- status
 - ehci.cpp, 8
- swap_points
 - ehci.cpp, 8
- swap_temp
 - ehci.cpp, 8
- update6dof
 - ehci.cpp, 6
- updateGlPositMatrix
 - ehci.cpp, 6
 - ehci.h, 12
- updateInternalHeadPosition
 - ehci.cpp, 7
- updateReferenceInternalHeadPosition
 - ehci.cpp, 7
- upperHeadCorner
 - ehci.cpp, 8
- win_size
 - ehci.cpp, 8