Reference Manual

Generated by Doxygen 1.5.6

Wed Aug 6 02:51:14 2008

Contents

Chapter 1

File Index

1.1 File List

Here is a list of all files with brief descriptions:

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 updateGlPositMatrix (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 updateGlPositMatrix (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 getGlPositMatrix (double myGlPositMatrix[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 documentation.

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 updateGlPositMatrix (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