



EyeMoShake Manual

EyeMoShake Version 1.0.0
Document Version 1.0.0

© 2006 GestureTek Inc.
317 Adelaide Street West, Toronto, Ontario, M5V 1P9 Canada
Web: <http://www.gesturetek.com> Email: support@gesturetek.com
Phone: 416.340.9290 Fax: 416.348.9809

Table of Contents

1	Introduction	1
1.1	Copyright and Legal Matters.....	1
1.2	Prerequisites.....	1
1.3	Device requirements.....	1
1.4	Overview.....	2
2	Methods	3
2.1	IEYEMOSHAKE_GetResult.....	3
2.2	IEYEMOSHAKE_GetResultRange.....	3
2.3	IEYEMOSHAKE_Initialize.....	4
2.4	IEYEMOSHAKE_ProcessFrame.....	5
2.5	IEYEMOSHAKE_Reset.....	6
3	Data Reference	8
3.1	EYEMOSHAKE_DATA_RAW_MOTION.....	8

1 Introduction

The EyeMoShake extension is an easy-to-use BREW interface that makes camera-based motion tracking available to BREW applications. The EyeMoShake extension uses a live stream of video frames captured from a device's camera to track raw motion of the device.

EyeMoShake could be used in applications that require motion detection, when precise measurements or direction are not needed. Possible applications include virtual dice or snow globe. For more advanced motion tracking requirements, EyeMoRocknRoll is also available.

It is recommended that EyeMoShake be used in conjunction with the EyeMoCamera extension. EyeMoCamera simplifies camera access by handling device-specific initialization and implementation complexities that may occur, such as devices that contain more than one camera, various camera orientations, devices supporting only a subset of common resolutions, and those devices which require unusual initialization steps. For more information regarding the EyeMoCamera extension, see the EyeMoCamera documentation.

1.1 Copyright and Legal Matters

GestureTek and EyeMobile are registered trademarks of GestureTek Inc.

BREW is a registered trademark of QUALCOMM Incorporated.

GestureTek technology and software is protected under United States patent number 5,534,917 (Video image based control system).

The contents of this documentation are the property of GestureTek Inc.

1.2 Prerequisites

Familiarity with BREW development in the C programming language.

1.3 Device requirements

- Device with BREW version 2.1.3 or above.
- Device camera should be available to BREW (device specifications are available from the BREW Developer extranet).
- Color or grey-scale camera images are supported.
- Frame rate must be at least 15 frames per second.
- Minimum supported resolution is 80x60 pixels.
- For best results the camera image should represent the entire field-of-view of the camera, rather than a sub-window of the original view.
- A small amount of memory is required for EyeMoCamera to allocate its internal buffers.
- BREW Simulator (included with BREW SDK version 3 and above) is required to run camera-enabled BREW applications directly from Microsoft Visual Studio.
- Running EyeMoShake-enabled applications on the BREW Simulator requires a webcam.

1.4 Overview

The EyeMoShake extension tracks device movement by determining the differences between consecutive frames.

The extension makes use of both texture and gradients in luminance when comparing images. This allows it to operate in a wide range of static environments, including those with few distinctive objects or features. Blurry images are also well detected, allowing the system to track fairly quick movements.

In order to utilize EyeMoShake, an application must do the following:

- Create a new instance of the extension using `ISHELL_CreateInstance` and pass it the classID `AEECLSID_EYEMOSHAKE`.
- Initialize the extension with the frame size and the image format of the incoming images using `IEYEMOSHAKE_Initialize`^[4].
- Pass frames into the extension using `IEYEMOSHAKE_ProcessFrame`^[5].
- Query the data that the extension has computed using `IEYEMOSHAKE_GetResult`^[3] for any of the data channels that the tracker supports.
- Reset the tracker as needed using `IEYEMOSHAKE_Reset`^[6].
- And finally, release the extension when it is no longer being used, with `IEYEMOSHAKE_Release`.

A number of the above steps are automated when using EyeMoCamera. The application no longer needs to call `IEYEMOSHAKE_Initialize`, or `IEYEMOSHAKE_ProcessFrame` after the EyeMoShake has been passed to `IEYEMOCAMERA_AddFrameListener`. For more information regarding the EyeMoCamera extension, see the EyeMoCamera documentation.

2 Methods

2.1 IEYEMOSHAKE_GetResult

Returns the current data value of the data stream associated with the given ID. Currently, only `EYEMOSHAKE_DATA_RAW_MOTION` is supported. This function can be called multiple times per frame, and the values returned by the function are only updated after each call to `IEYEMOSHAKE_ProcessFrame`.

Definition

```
int
IEYEMOSHAKE_GetResult(
    IEyeMoShake * po,
    int32 nDataID,
    void * pValue
);
```

Parameters

`po`

Pointer to an instance of the extension interface object.

`nDataID`

The ID of the data stream that is to be queried.

`pValue` (*Output*)

A pointer to the variable that receives the returned data value. It is the application's responsibility to allocate a structure of the appropriate size for this function to store the return value.

Returns

`EYEMO_OKAY`

The tracker has returned the data without warnings.

`EYEMO_NOT_SUPPORTED`

The ID of the data stream given by `nDataID` is invalid.

`EYEMO_ERROR_NOT_INITIALIZED`

The extension has not yet been initialized.

`EYEMO_ERROR`

The tracker has generated an error.

`EYEMO_WARNING_NO_DATA`

The tracker has not yet generated any motion data. This is returned on the first few frames, until the tracker has analyzed sufficient information to generate motion data.

Notes

None

2.2 IEYEMOSHAKE_GetResultRange

Returns the usable range of the given data stream. If the range of the given data stream is dependent on the current parameters or initialization settings of the extension, the values returned by this function will always contain the correct range for the current settings. If the settings changes later, another call to `IEYEMOSHAKE_GetResultRange` will be necessary.

Definition

```

int
IEYEMOSHAKE_GetResultRange(
    IEyeMoShake * po,
    int32 nDataID,
    void * pMin,
    void * pMax
);

```

Parameters

po

Pointer to an instance of the extension interface object.

nParamID

The ID of the parameter that is to be queried. In EyeMoShake, only possible value is EYEMOSHAKE_DATA_RAW_MOTION.

pMin (*Output*)

A pointer to the memory location where the minimum value of the data stream will be stored. It is the application's responsibility to allocate the appropriate amount of space to store the return value.

pMax (*Output*)

A pointer to the memory location where the maximum value of the data stream will be stored. It is the application's responsibility to allocate the appropriate amount of space to store the return value.

Returns

EYEMO_OKAY

The tracker has returned the parameter without warnings.

EYEMO_NO_RANGE

The current data stream does not have a meaningful range associated with it. This can occur when the type of the data stream does not have a meaningful order defined on it, as is the case with a structure or an enumerated type.

EYEMO_NOT_SUPPORTED

The ID of the data stream given by nParamID is invalid.

EYEMO_ERROR_NOT_INITIALIZED

The extension has not yet been initialized.

EYEMO_ERROR

The tracker has generated an error.

Notes

None

2.3 IEYEMOSHAKE_Initialize

Initializes the EyeMoShake extension. This function allocates the necessary buffers and initializes the underlying tracker.

Definition

```

int
IEYEMOSHAKE_Initialize(
    IEyeMoShake * po,
    uint16 nFrameW,
    uint16 nFrameH,
    int32 nColorMode
);

```

Parameters

`po`

Pointer to an instance of the extension interface object. This object must be allocated separately using `ISHELL_CreateInstance` before calling this function. Use the constant `AEECLSID_EYEMOSHAKE` as the `CLSID` value for the instance.

`nFrameW, nFrameH`

The dimensions of the video frames that will be passed to the extension for processing.

`nColorMode`

The color mode of the incoming video frames.

Returns

`EYEMO_OKAY`

No problems were encountered during initialization.

`EYEMO_ERROR_NO_MEMORY`

There was insufficient memory available to allocate the necessary resources.

`EYEMO_ERROR`

One of the components could not be initialized.

`EYEMO_WARNING_ALREADY_INITIALIZED`

The extension was already initialized.

Notes

The `nFrameW` and `nFrameH` parameters define the dimensions of the video frames that will be passed to the `IEYEMOSHAKE_ProcessFrame` function. Due to the inconsistencies between BREW implementations on different devices, obtaining an accurate frame size prior to starting the camera is not very reliable. The most accurate way of finding out the actual frame size is to start the camera and wait until you receive the first frame. Then you can use the dimensions of this frame as values for these parameters. However, if you are designing for a small subset of devices and can reliably predict the camera frame size, you can call this function prior to starting the camera without waiting for the first frame.

The `nColorMode` will need to be set to match the color format that the device is using. Since initialization is usually deferred until the first frame of video is received, the color mode can be determined by casting the frame to an `IDIB`, and retrieving the `nColorScheme` field. EyeMoShake supports a number of color modes, but the value used will typically be one of the following (which are simply aliases to the appropriate platform enums):

```
EYEMO_COLORMODE_X1R5G5B5
EYEMO_COLORMODE_R5G6B5
EYEMO_COLORMODE_X14R6G6B6
EYEMO_COLORMODE_R8G8B8
```

2.4 IEYEMOSHAKE_ProcessFrame

Called whenever a new video frame is captured. The calculated results are stored by the extension and can be retrieved by the application using `IEYEMOSHAKE_GetResult`.

Definition

```
int
IEYEMOSHAKE_ProcessFrame(
    IEyeMoShake * po,
    IBitmap * pFrame,
    uint32 nElapsedTicks
);
```

Parameters

`po`

Pointer to an instance of the extension interface object.

pFrame

A pointer to the captured frame. You can pass the same pointer returned by the camera callback function.

nElapsedTicks

The number of milliseconds elapsed since the last frame was produced by the camera.

Returns

EYEMO_OKAY

The tracker has processed the motion without warnings.

EYEMO_WARNING_NO_DATA

The tracker has not yet generated any motion data. This is returned on the first few frames, until the tracker has analyzed sufficient information to generate motion data.

EYEMO_WARNING_STARTUP

This warning is generated whenever the tracker is still in the process of starting up. When the device camera first starts, the first few frames captured are not very reliable; therefore, the results generated by the tracker will not be reliable. During this time, results will be generated, but may be less accurate than normal.

EYEMO_WARNING_LOW_DETAIL

This warning is generated whenever there is very little detail within the image, such as when the camera is pointing at a nearly blank wall or floor. Motion will still be estimated, though the calculated results may be less reliable. This threshold can be set by the Low Detail parameter.

EYEMO_WARNING_TOO_FAST

This warning is generated whenever consecutive frames do not appear to have any overlap. The most likely cause being that the device was moved too fast. Motion cannot be determined if there is no overlap, and the motion results will be zero when this result is returned.

EYEMO_ERROR_PAUSED

This warning is generated if the function is called while the tracker is paused.

EYEMO_ERROR_INVALID_IMAGE

This warning is generated when the input frame is invalid. This can occur if the pointer is NULL, or if the color mode or the dimensions of the frame do not match the parameters used to initialize the tracker.

EYEMO_ERROR

The tracker has generated an error.

Notes

None

2.5 IEYEMOSHAKE_Reset

Recenter the tracker.

Definition

```
int
IEYEMOSHAKE_Reset(
    IEyeMoShake * po
);
```

Parameters

po

Pointer to an instance of the extension interface object.

Returns

EYEMO_OKAY

The tracker was successfully reset.

EYEMO_ERROR_NOT_INITIALIZED

The extension has not yet been initialized.

EYEMO_ERROR

An unexpected error has occurred.

Notes

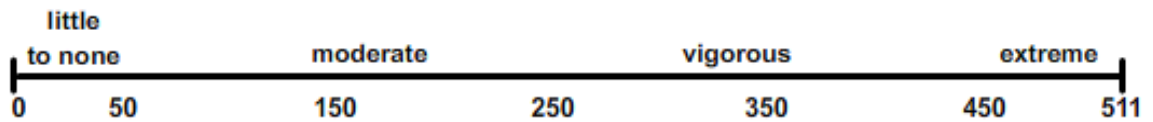
Resetting the tracker sets all tracking values to 0. The application should recenter the tracker at the start of a game or level, as well as allowing the user to press a key to recenter the device so that the tracker can be reset when the position of the device has been changed, such as when the device is handed to another person.

3 Data Reference

3.1 EYEMOSHAKE_DATA_RAW_MOTION

The IEyeMoShake extension computes a value that represents the amount of motion between consecutive camera frames. The motion measurement is corrected for the scene complexity and lighting. Therefore, the value will be approximately independent of the scene that the camera is viewing. It is also independent of the resolution of the images that are processed.

The motion measurement has a range of 0 to 511, inclusive. The magnitude correlates to the type of motion to which the camera is subjected as illustrated here:



The motion measurement may be used by the application to differentiate between gentle and vigorous shaking motions. Alternatively, the application may apply a threshold to the motion measurement, if the application requires only motion state or still state information.

Type

Integer (*int32*)

Range

0–511 (*inclusive*)