Technical Documentation

**Table 1. Document Change Control**

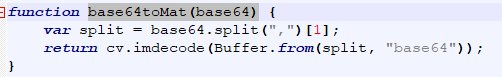
|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Authors** | **Summary of Changes** |
| 1.0 | 14/10/2018 | Shenal Samarasinghe  Dineth Gunawardena | Initial draft created |
| 1.1 | 14/10/2018 | Shenal Samarasinghe  Dineth Gunawardena  Ayub Khan | Data was added to certain sections |
| 1.2 | 28/10/2018 | Lyndon | Updating makeHandMask, constants, getHandContour,getObjectCentre,getTraceCoordinates, calibrateHSV, log |

# 

# FrameTrace.js

The of the this class has been deprecated and transferred to the client side

## base64toMat(base64)



**Definition:**Converts a base64 value to a Mat value

**Parameter:**Base64

**Return:**Mat

# Gesture.js

This file has the code that is used to extract the hand from the frame

## Variable

**lH-** Low Hue

**lS-** Low Saturation

**lV-** Low Value

**uH-** Upper Hue

**uS-** Upper Saturation

**uV-** Upper Value

**hueVariance-** Acceptable range of Hue

**satVariance-** Acceptable range of Saturation

**valVariance-** Acceptable range of Value

## Constant

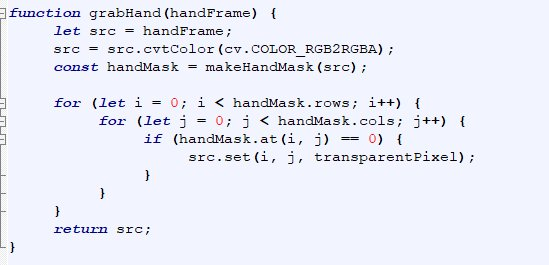
**transparentPixel**

## grabHand(handFrame)

**Definition:** Extracts hands from frame using makeHandMask()

**Parameter:** *Mat* handFrame

**Return:** *Mat* with the original frame with inverted hand mask

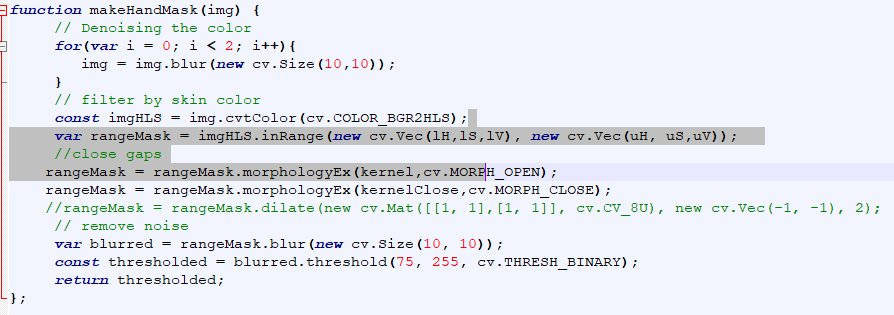


## makeHandMask(img)

**Definition:** Create a Mat Frame that consists of bits with values of 0s or 1s to signify the threshold of the color that is selected i.e. if the raw image pixel is the selected color, return a 1 else if not the color return a 0. This function generates the mask for one frame to be used for hand extraction.

**Parameter:** Mat of Vec3 RGB values

**Return:** Mat of Binary values

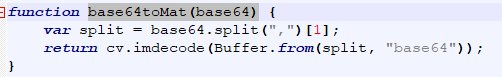


## base64toMat(base64)

**Definition:** Splits base64 string and converts it to Mat

**Parameter:** *String* base64 string

**Return:** *Mat*

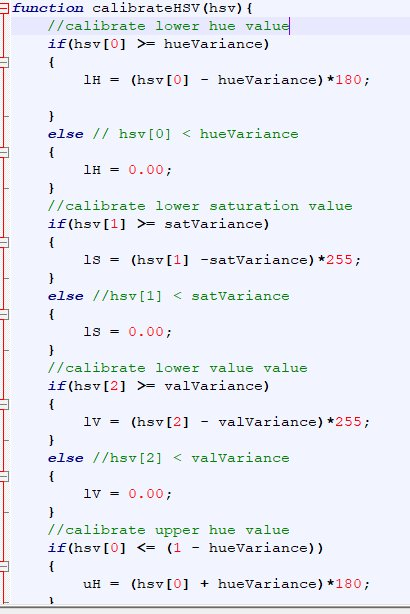


## calibrateHSV(hsv)

**Definition:** Calibrate upper and lower HSV values

**Parameter:** Float [3]

**Return:** void



# GetTraceCoordinates.js

This file has the code needed to get the coordinates of the center of the pointer

## Variables

**lH-** Low Hue

**lS-** Low Saturation

**lV-** Low Value

**uH-** Upper Hue

**uS-** Upper Saturation

**uV-** Upper Value

**hueVariance-** Acceptable range of Hue

**satVariance-** Acceptable range of Saturation

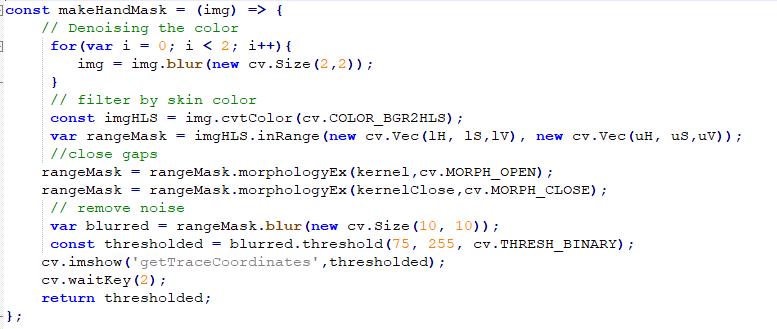
**valVariance-** Acceptable range of Value

## Constants

kernel = Morphology opening - Deprecated no longer used (used for dilation and erosion)

kernelClose = Morphology closing - Deprecated no longer used (used for dilation and erosion)

## makeHandMask(img)

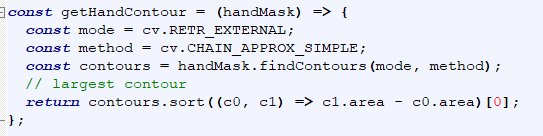


**Definition:**Create a Mat Frame that consists of bits with values of 0s or 1s to signify the threshold of the color that is selected i.e. if the raw image pixel is the selected color, return a 1 else if not the color return a 0. This function generates the mask for one frame to be used for hand extraction.

**Parameter:** Mat of Vec3 RGB values

**Return:** Mat of Binary values

## getHandContour (handMask)

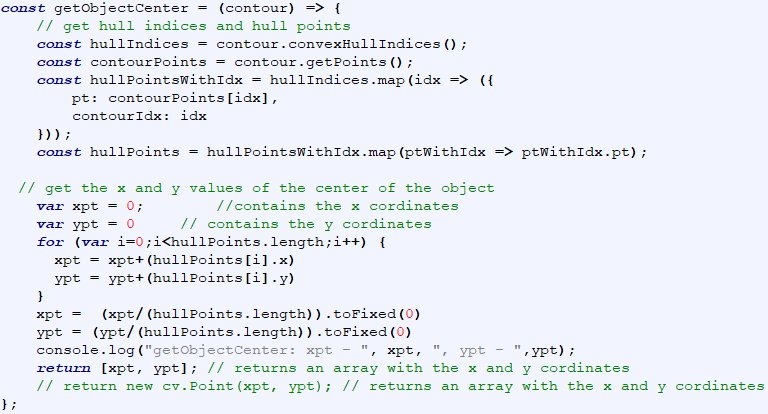


**Definition:**Uses the hand mask to get the hand contour

**Parameter:** Mat of Binary values

**Return:** Contour[]

## getObjectCenter(contour)

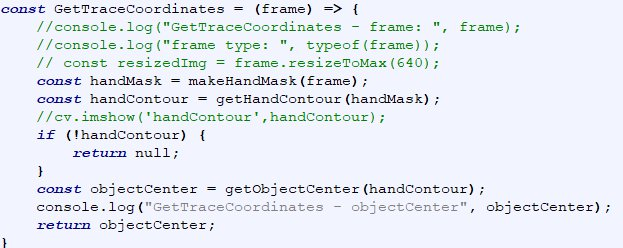


**Definition:**It takes the hull points from the contour and uses the hull points to calculate the center points of the pointer.

**Parameter:**Contour[]

**Return:**Point2d

## GetTraceCoordinates(frame)

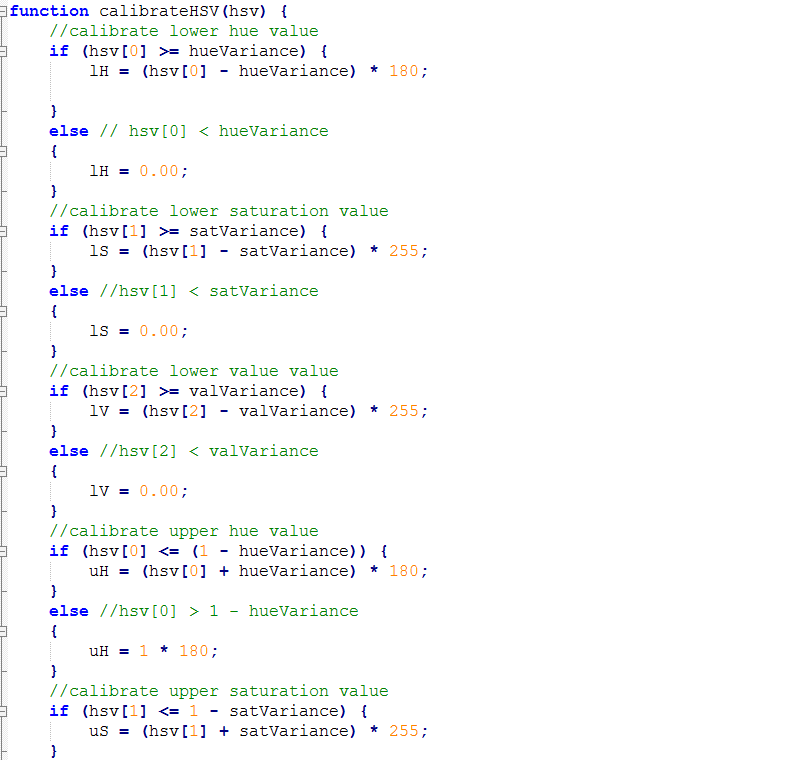


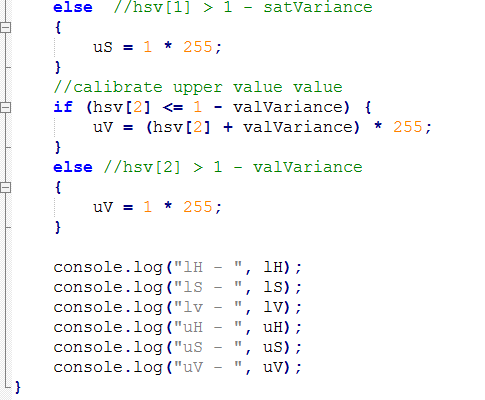
**Definition:**It acts as the main function,

**Parameter:** Mat (Raw image)

**Return:**Point2d

## calibrateHSV(hsv)





**Definition:**It calibrates the HSV values

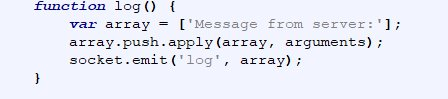
**Parameter:**float[3]

**Return:**Void

# Utils.js

# webRTC.js

## log()



**Definition:** emits message from server

**Parameter:** void

**Return:** void