adaptive-motion v0.2.0 build passing

This module allows you to control an AngularJS app using web camera.

Demo

Check out http://angular-adaptive.github.io/adaptive-motion/demo/

Requirements

- AngularJS v 1.0+
- getUserMedia Stream support

Usage

We use bower for dependency management. Add

```
dependencies: {
    "angular-adaptive-motion": "latest"
}
```

To your `bower.json` file. Then run

```
bower install
```

This will copy the angular-adaptive-motion files into your `bower_components` folder, along with its dependencies. Load the script files in your application:

```
<script type="text/javascript" src="bower_components/angular/angular.js"></script>
<script type="text/javascript"
    src="bower_components/angular-adaptive-motion/angular-adaptive-motion.js"></script>
```

Add the **adaptive.motion** module as a dependency to your application module:

```
var myAppModule = angular.module('MyApp', ['adaptive.motion']);
```

and include `$motion` service as a dependency to your controller:

```
angular.module('MyApp').controller('MainCtrl',
    function ['$scope', '$motion', ($scope, $motion) {

}]);
```

Public methods

$motion.start();

Starts gesture recognition.

$motion.stop();

Stops gesture recognition.

$motion.onStart(cb);

On start callback.

$motion.onStop(cb);

On stop callback.

$motion.onError(cb);

On error callback.

$motion.onSwipeLeft(cb);

On swipe left gesture.

```
$motion.onSwipeLeft(function(data){
    $scope.$apply(function(){
        console.log('onSwipeLeft');
    });
});
```

$motion.onSwipeRight(cb);

On swipe right gesture.

```
$motion.onSwipeRight(function(data){
    $scope.$apply(function(){
        console.log('onSwipeRight');
    });
});
```

$motion.onSwipeUp(cb);

On swipe up gesture.

```
$motion.onSwipeUp(function(data){
    $scope.$apply(function(){
        console.log('onSwipeUp');
    });
});
```

$motion.onSwipeDown(cb);

On swipe down gesture.

```
$motion.onSwipeDown(function(data){
    $scope.$apply(function(){
        console.log('onSwipeDown');
    });
});
```

Configuration

You can configure `$motionProvider` to a custom treshold options in app configuration.

```
$motionProvider.setTreshold({
    'rgb': 150,
    'move': 3,
    'bright': 300
});
```

You can also set custom hsv filter.

```
$motionProvider.setHsvFilter({
    'huemin': 0.0,
    'huemax': 0.1,
    'satmin': 0.0,
    'satmax': 1.0,
    'valmin': 0.4,
    'valmax': 1.0
});
```

Visualization

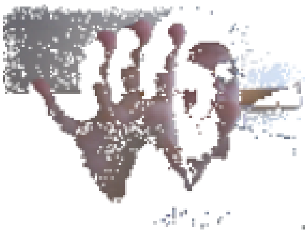
If you want to visualize you can add `adaptive-motion` attribute into your canvas element. You can choose from following styles:

Video

```
<canvas adaptive-motion="video"></canvas>
```

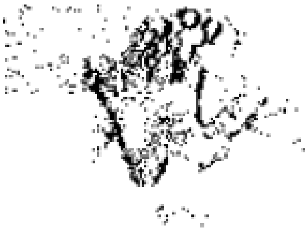
Skin

```
<canvas adaptive-motion="skin"></canvas>
```



Edge

```
<canvas adaptive-motion="edge"></canvas>
```



Contributing

Contributions are welcome. Please make a pull request against canary branch and do not bump versions. Also include tests.

Testing

We use karma and jshint to ensure the quality of the code. The easiest way to run these checks is to use grunt:

```
npm install -g grunt-cli
npm install
bower install
grunt
```

The karma task will try to open Chrome as a browser in which to run the tests. Make sure this is available or change the configuration in `test/test.config.js`