

# Text to Motion Database

## Test Plan

Brendan Duke  
Andrew Kohnen  
Udip Patel  
David Pitkanen  
Jordan Viveiros

November 1, 2016

# Contents

<b>1</b>	<b>Overview</b>	<b>1</b>
1.1	Test Case Format . . . . .	1
1.2	Automated Testing . . . . .	1
1.2.1	Testing Tools . . . . .	1
1.3	Manual Testing . . . . .	1
1.3.1	User Experience Testing . . . . .	1
1.4	List of Constants . . . . .	1
<b>2</b>	<b>Proof of Concept Testing</b>	<b>2</b>
2.1	Significant Risks . . . . .	2
2.2	Demonstration Plan . . . . .	2
2.3	Proof of Concept Test . . . . .	2
<b>3</b>	<b>System Testing</b>	<b>3</b>
<b>4</b>	<b>Constraints Testing</b>	<b>4</b>
4.1	Solution Constraints Testing . . . . .	4
<b>5</b>	<b>Functional Requirements Testing</b>	<b>6</b>
<b>6</b>	<b>Non-Functional Requirements Testing</b>	<b>7</b>
6.1	Look and Feel Requirements Testing . . . . .	7
<b>7</b>	<b>Timeline</b>	<b>8</b>
<b>8</b>	<b>Appendix A: Testing Survey</b>	<b>9</b>

## List of Tables

# List of Figures

## Revision History

Date	Version	Notes
October 25, 2015	1.0	Created document
October 31, 2015	1.1	Major additions to all sections
November 1, 2015	1.2	Final version for rev 0

# **1 Overview**

## **1.1 Test Case Format**

## **1.2 Automated Testing**

### **1.2.1 Testing Tools**

## **1.3 Manual Testing**

### **1.3.1 User Experience Testing**

## **1.4 List of Constants**

## **2 Proof of Concept Testing**

### **2.1 Significant Risks**

### **2.2 Demonstration Plan**

### **2.3 Proof of Concept Test**

## **3 System Testing**

## 4 Constraints Testing

### 4.1 Solution Constraints Testing

<b>Test 4.1.1:</b>	<b>Deep Learning Methods Test</b>
<b>Description:</b>	Test whether the human pose estimation component of the software uses modern deep learning methods.
<b>Type:</b>	Manual
<b>Testers:</b>	Supervisor (Dr. Taylor)
<b>Pass:</b>	Dr. Taylor should confirm that the deep learning methods used are satisfactory and relevant to current research, with a yes or no response.
<b>Req. #:</b>	1

<b>Test 4.1.2:</b>	<b>Standard Data Format Test</b>
<b>Description:</b>	Tests whether the human pose data format used in the project is standard, and compatible with existing software libraries.
<b>Type:</b>	Automated
<b>Initial State:</b>	Initialize database query interface.
<b>Input:</b>	Random ID of a tuple, containing human pose data, in the database.
<b>Output:</b>	Tuple containing data in HDF5 format.
<b>Pass:</b>	The human pose datum should be parseable by an existing HDF5 data library.
<b>Req. #:</b>	2



<b>Test 4.1.3:</b>	<b>Linux Platform Build and Run Test</b>
<b>Description:</b>	Confirms that all nightly build tests, as well as the automated test suite, are working under Linux.
<b>Type:</b>	Automated
<b>Initial State:</b>	None (build test).
<b>Input:</b>	Commands to begin build and run sequence.
<b>Output:</b>	Compile and run success, or errors.
<b>Pass:</b>	Compile and run success.
<b>Req. #:</b>	3

<b>Test 4.1.4:</b>	<b>Python API Hook Testing</b>
<b>Description:</b>	Confirms that major module interfaces, such as the image pose estimation interface, and database query interface, have working Python hooks.
<b>Type:</b>	Automated
<b>Initial State:</b>	Initialization specific to each module interface under test.
<b>Input:</b>	Valid parameters for each module interface, written in Python.
<b>Output:</b>	Expected success-case outputs for each module interface, written in Python.
<b>Pass:</b>	Interface calls completed without error, and returned their expected outputs.
<b>Req. #:</b>	4

## 5 Functional Requirements Testing

<b>Test 5.1:</b>	<b>Supported Video Encodings Test</b>
<b>Description:</b>	Tests whether the ReadFrames API is able to decode MP4, MP2 and AAC video files.
<b>Type:</b>	Automated
<b>Initial State:</b>	Call read frames initialization procedure.
<b>Input:</b>	30 second MP4 video file at 30 FPS.
<b>Output:</b>	A set of 900 ( $30 \times 30$ ) frames.
<b>Pass:</b>	The 900 frames match a set of 900 expected frames from a reference frame-reading system.
<b>Req. #:</b>	7

<b>Test 5.2:</b>	<b>Frame Reading Timestamp Accuracy Test</b>
<b>Description:</b>	Tests whether the timestamps on the frames returned by the ReadFrames API match their temporal position in the original video stream.
<b>Type:</b>	Automated
<b>Initial State:</b>	Call read frames initialization procedure.
<b>Input:</b>	30 second MP4 video file at 30 FPS.
<b>Output:</b>	A set of 900 ( $30 \times 30$ ) frames, which include time stamps.
<b>Pass:</b>	The timestamps on the 900 frames match a set of timestamps on a test vector of expected timestamps for the 900 frames.
<b>Req. #:</b>	8

## 6 Non-Functional Requirements Testing

### 6.1 Look and Feel Requirements Testing

<b>Test 6.1.1:</b>	Colour Scheme Test
<b>Description:</b>	Test user satisfaction of the web interface colour scheme.
<b>Type:</b>	Manual
<b>Testers:</b>	Testing Group
<b>Pass:</b>	On a one to ten scale, the average user rating is above six.
<b>Req. #:</b>	12

## 7 Timeline

## **8 Appendix A: Testing Survey**