Text to Motion Database

Test Plan

Brendan Duke
Andrew Kohnen
Udip Patel
David Pitkanen
Jordan Viveiros

November 1, 2016

Contents

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

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

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

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

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

5 Functional Requirements Testing

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

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

Test 5.3:	**Video Human Pose Estimation Data Quality Test**
Description:	Test to ensure the data quality produced by the human pose estimator component. A set of Charades videos will be processed by the human pose estimator, and skeleton animations corresponding to the generated human pose data will be created (this is a scoped part of the software pipeline). A double-blind test will be ran, where testers will be shown random mixed sets of the skeleton animations produced by McMaster Text to Motion, together with skeletons from actual motion capture data coming from CMU's motion capture lab. Testers will indicate whether they think the motion capture data came from actual motion capture, or from the pose estimation software.
Type:	Manual
Testers:	Testing Group
Pass:	Within a 5% confidence interval, the McMaster Text to Motion skeletons will be indicated as being actual motion capture data with the same probability that the CMU motion capture skeletons are indicated as being actual motion capture data.
Req. #:	8

Test 5.4:	**Database Output Full Range Coverage Test**
Description:	Tests whether the range of the text-to-motion database search is equal to the entire set of data stored in the database.
Type:	Automated
Initial State:	Initialize database-query and full text search module interfaces. Populate database with Charades data.
Input:	A random matching keyword from the text description of each video (acquired automatically).
Output:	A set of video-pose data from the database that should include the original datum that the input keyword was taken from.
Pass:	The returned set of data contains the original video record.
Req. #:	9

Test 5.5:	**Database No False Positives Test**
Description:	Tests whether the results retried from text searches of the database contain any false positives, i.e. results whose text descriptions do not contain any of the searched keywords.
Type:	Automated
Initial State:	Initialize database-query and full text search module interfaces. Populate database with Charades data.
Input:	For each video, a random set of keywords not in that video's text description.
Output:	A set of video-pose entries.
Pass:	The output set of data should not contain the original video that was chosen to be outside the subset of the output range for this input.
Req. #:	10

Test 5.6:	**Full Text Search Order by Relevance Test**
Description:	A test of whether the full text search interface is returning a set of entries that are ordered by relevance to the search keywords.
Type:	Automated
Initial State:	Initialize database-query and full text search module interfaces. Populate database with Charades data.
Input:	A random set of search keywords, drawn automatically from the set of text descriptions in the database.
Output:	A set of entries in the database, in some order.
Pass:	The output set of entries should be randomly ordered and input to a reference full text search engine, which will produce an expecteed ordering by relevance. A statistical test of the similarity of the McMaster Text to Motion ordering and the reference ordering should be done, and the McMaster Text to Motion ordering should be expected to be the same within a 5% confidence interval.
Req. #:	11

6 Non-Functional Requirements Testing

6.1 Look and Feel Requirements Testing

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

7 Timeline

8 Appendix A: Testing Survey