

## Dataset // Dataset Viewer // Supplementary PDF

This document supplements the Dataset and the “Viewer” released in the frame of the GCPR paper “A Comparison of Directional Distances for Hand Pose Estimation”. It gives a short introduction on the use of the viewer-application and the structure of the dataset and its use.

The files of the viewer released contain both an 64-bit linux executable and the source code (QT project - one can use the standard QT tools to open the project. Only Ubuntu OS is supported, with all the needed libraries installed through packages). In the case of a different OS or custom library installations the user should do some changes in the .pro file, regarding the libraries, and compile the source code. Regarding the rest of the code, only one button (which just opens a path in a window) is OS-specific and the Linux-command used should be replaced with an appropriate OS-specific one. (this is noted later on, at the corresponding section)

The project is a QT project, using QT Creator 2.5.0 and QT 4.8.1. The OS used is Ubuntu 12.04 64 bit.

The dependences are:

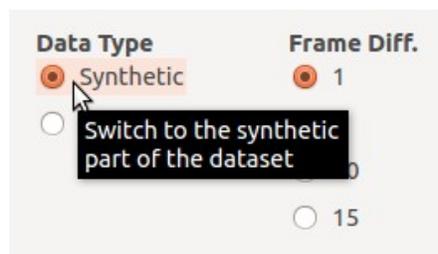
- QT libraries and tools (sudo apt-get install libqt4-dev qt4-dev-tools qtcreator)
- Eigen library (sudo apt-get install libeigen3-dev) - <http://eigen.tuxfamily.org>
- OpenCV (sudo apt-get install libopencv-dev) - <http://opencv.willowgarage.com>
- GPU with OpenGL support.

In case of comments/question, please send an email to [dimitris.tzionas@tuebingen.mpg.de](mailto:dimitris.tzionas@tuebingen.mpg.de)

---

### PART A - Dataset Viewer

In general, the application is expected to be pretty self-explainable, as it doesn't have complex parameter inputs, and guiding tooltips are used for most controls/widgets on mouse hover, as shown in the following image.

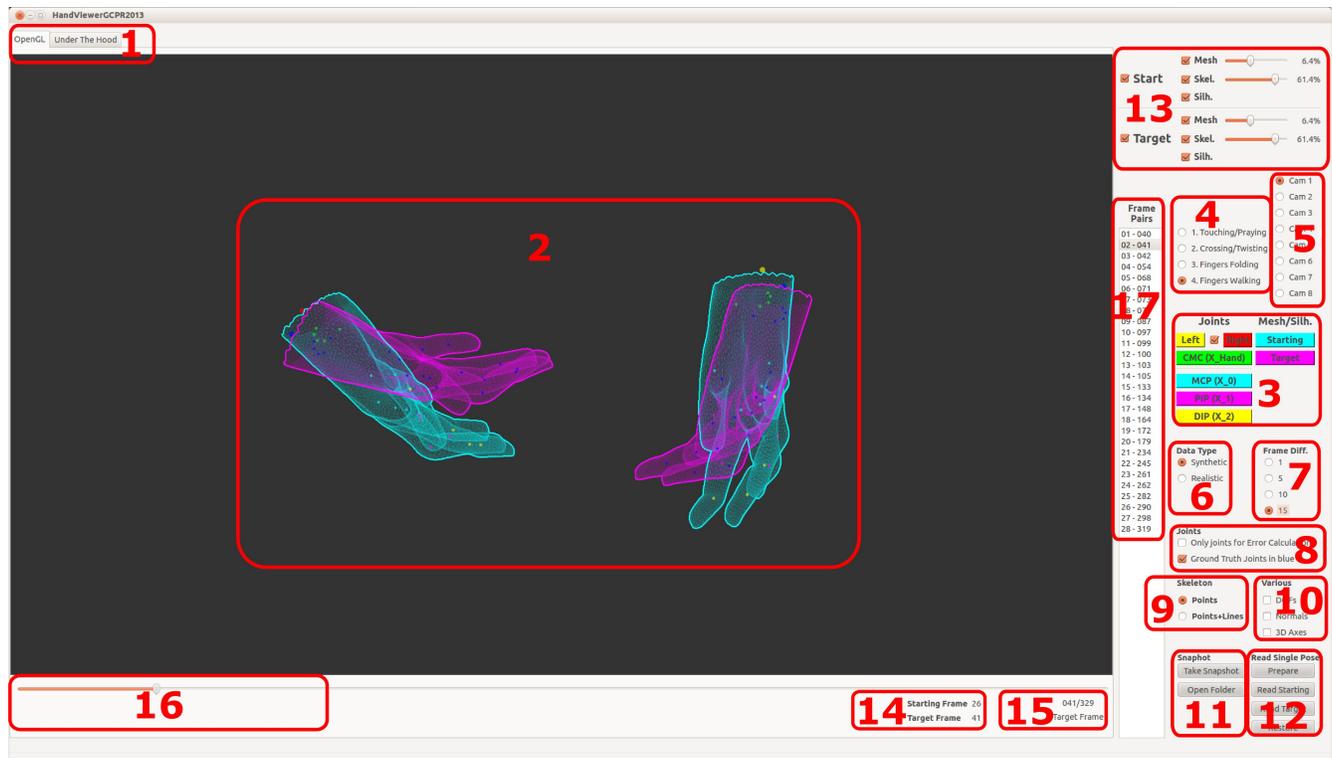


The main control of the application is located at the upper left corner of the window, and changes the tab depicted in the window. There are 2 tabs with different functionalities, as described in the following.



## Tab - OpenGL

This is the main tab of the application and is the actual viewer. In this case the window looks like the following image, where several function-areas are annotated and explained in the following part of the section.



1. The tab widget, which was described in the previous section. It toggles between the actual viewer and some supplementary widgets that give a more in depth intuition about the hand model.
2. The OpenGL widget where the 3D scene is projected. When the synthetic part of the dataset is selected (area 6), the 3D scene can be rotated using the right and left mouse buttons and by moving the mouse. A helping set of 3D coordinate system axes (Red=X, Green=Y, Blue=Z) can be added in the scene through a check-box in area 10, that helps the user familiarize himself with the rotation around all 3 axes.
  - Left click + Vertical movement = Rotation around X (red)
  - Left click + Horizontal movement = Rotation around Y (green)
  - Right click + Vertical movement = Rotation around X (red)
  - Right click + Horizontal movement = Rotation around Z (blue)
3. The buttons responsible to change the colors of all the parts/models of the 3D scene. Each button and its corresponding 3D part/model have the same color, thus the color of the buttons can be used to identify the 3D parts/models of the scene.
4. The radio buttons responsible to change the (video/motion) sequence depicted. This can also be done though the *keyboard*, by pressing the key “ctrl” and one of the keys 1-4.
5. The radio buttons responsible to change the camera view, thus the projection of the 3D scene. This can also be done though the *keyboard*, using the keys 1-8.
6. The radio buttons responsible to toggle between the synthetic and the realistic part of the dataset.

7. The radio buttons responsible to change the frame difference between the starting and the target pose (testing frame) of the depicted pair of poses/frames. This can also be done though the *keyboard*, by pressing the key “alt” and one of the keys 1-4.
8. Options regarding the joints depicted.
9. Options regarding the rendering form of the skeleton.
10. Some extra rendering options.
11. A tool to take snapshots of the 3D scene as depicted/projected in the OpenGL widget. The button “Open Folder” is the one that uses an OS specific<sup>1</sup> command (*nautilus path*) to open the path containing all the snapshots in a window. Its code is in the file “mainwindow\_GUI.cpp” in the method “MainWindow::on\_myButton\_OpenPath\_Output\_SingleFrames\_clicked()”
12. An example of how to read a single pose from the files of the dataset (and not the full sequence containing multiple poses, used originally). *This has to be used in the following way*: First, press the “Prepare” button, to set the appropriate rendering-parameters. Then, iterate as wished between the “Read Starting” and “Read Target” buttons, that read the starting and target pose respectively, and depict the result. The code behind these 2 buttons can be found in the file “mainwindow\_EXPERIMENT\_CLICK.cpp”, in the methods
  - “MainWindow::on\_myButton\_EXPERIMENT\_TestPose\_INI\_clicked()” and
  - “MainWindow::on\_myButton\_EXPERIMENT\_TestPose\_FIN\_clicked()”.
 At the end, before going back to the normal usage, the user must press the button “Restore”, to restore the default rendering-options, otherwise some unexpected results might be observed. An alternative would be to restart the application.
13. Rendering controls for the visibility of the models/parts of the 3D scene depicted in the OpenGL widget. The check-boxes make the respective models/parts appear/disappear, while the sliders change their opacity.
14. The indexes in the original sequence of the Starting and Target (testing) Frame/Pose of the selected pair.
15. The index of the Target (testing) Frame/Pose of the selected pair, and the full length of the original sequence.
16. The (non-interactive!) slider shows in a graphical way the position of the Target (testing) Frame/Pose of the selected pair, and the full length of the original sequence.
17. The listbox containing all the frame/pose couples for the selected sequence. The couples are created by sampling the target (testing) frames from the original sequence (by Ballan et al.) and then defining the starting pose according to the frame difference chosen in the area 7.

---

<sup>1</sup> as described in the first section of the text

## Tab - Under The Hood

This tab doesn't contain any functional part of the viewer, but is provided with some tools that give a more in-depth intuition about the hand model used (released by Ballan et al. - <http://cvg.ethz.ch/research/ih-mocap/>)

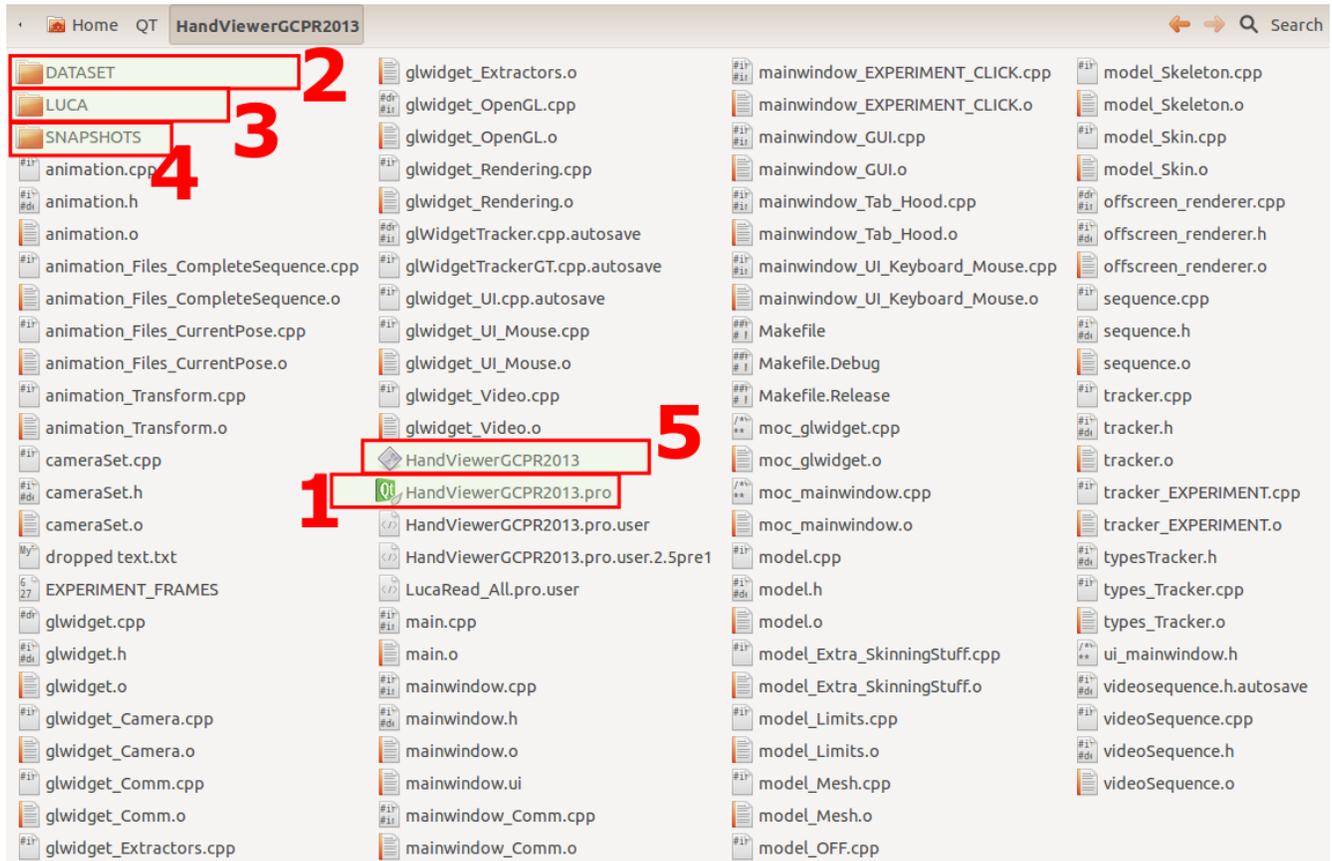
The screenshot shows the HandViewerG CPR2013 application interface. Red boxes and numbers 1-8 highlight the following features:

- 1:** The 'OpenGL Under The Hood' tab widget.
- 2:** The camera selection buttons (Cam 1 to Cam 8).
- 3:** The 'Camera 1' parameters table showing intrinsic and extrinsic camera data.
- 4:** The 'Model Info' table showing statistics for Vertices, Triplets, Bones, Skinning Bones, and Frames.
- 5:** The 'Vertices' table listing 3D coordinates for all vertices.
- 6:** The 'Triplets' table listing the indexes of the 3 corners of each triangle.
- 7:** The 'Bones' listbox showing a hierarchy of bones with columns for ID, Name, Length, Father, and Child.
- 8:** The 'Skinning Bones' listbox and the 'Skinning Weights' table, which includes a checkbox to show skinning weights for all vertices.

1. The tab widget, as described in the previous sections.
  2. The buttons responsible to change the camera view. This can be done also though the *keyboard*, using the keys 1-8.
  3. The camera parameters (intrinsic/extrinsic) of the selected camera.
  4. Some statistics of the hand model used and the sequence selected.
  5. The 3D coordinates of all the vertices.
  6. The indexes of the 3 corners of each triangle (triplet) of the mesh.
  7. The bones of the skeleton. On selection of a bone from the listbox, one can see at the right side of the area
    - The ID of the bone, its name and its length (right side - middle)
    - The ID and the name of its father in the kinematic chain (right side – up)
    - The number of its children in the kinematic chain, and a list of them. By single-clicking a child, one can see its ID in the original list. By double clicking on it, this bone is selected in the original listbox, and all the informations of this area are updated accordingly for this bone.
- One can climb down all the paths of the kinematic chain and have a better intuition about it.
8. A listbox containing all the skinning bones. Optionally (check checkbox) one can see the skinning weights of all the vertices for this skinning bone (notice that this is a slow process!)

## B - DATASET

The following image shows the main folder of the code-project released.



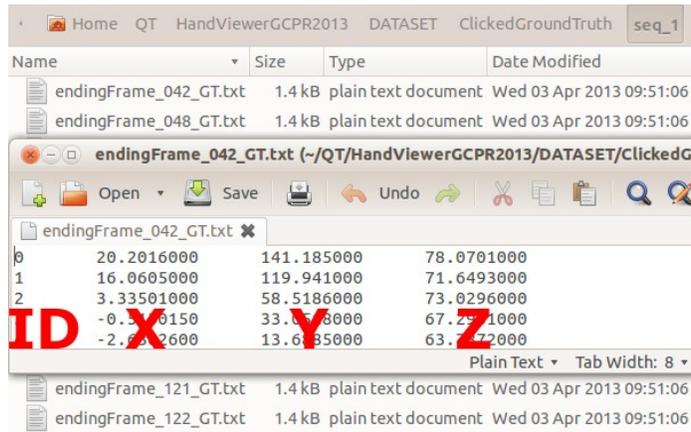
1. The QT project file. It can be opened with QT-Creator.
2. The folder of the dataset. See the following sections for details.
3. The folder containing files originally released by Ballan et al. (for the whole sequences - <http://cvg.ethz.ch/research/ih-mocap/>). See the last section for some helping information on how to read these files.
4. The folders containing the snapshots taken through the viewer-application.
5. Executable (Ubuntu 12.04, 64 bit, all libraries installed through packages at default locations)

### Folder - DATASET



1. Contains the manually annotated Ground Truth data. There is a .txt file for all the sampled ending-frames of all sequences.
2. Contains the files for the Realistic part of the dataset. There are 4 folders, for 1,5,10,15 frames difference respectively.
3. Contains the files for the Synthetic part of the dataset. There are 4 folders, for 1,5,10,15 frames difference

## Manual Ground Truth



For every sampled ending frame there exists one .txt file containing the corresponding ground truth joint positions.

These .txt files have the form depicted in the left image.

The number of lines is equal to the number of the skinning bones (40).

The first column contains the IDs of the skinning bones (0..39), while the other columns contain the 3D coordinates X, Y, Z.

Example code on how to read the files can be found in the file “tracker\_EXPERIMENT.cpp”, in the method “Tracker::Experiment\_ReadJointLocations\_GroundTruth\_CLICKED()”

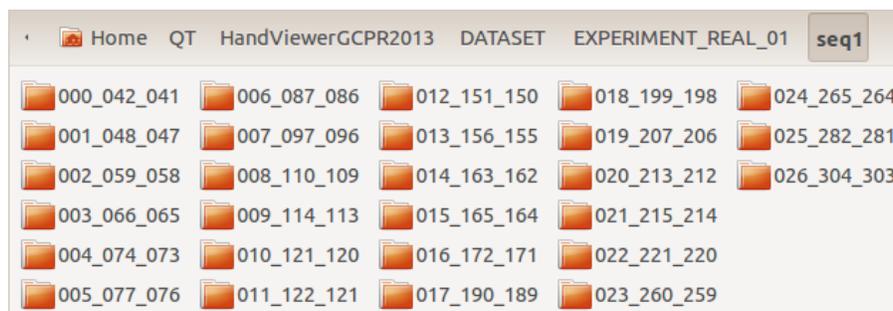
---

## Realistic/Synthetic parts

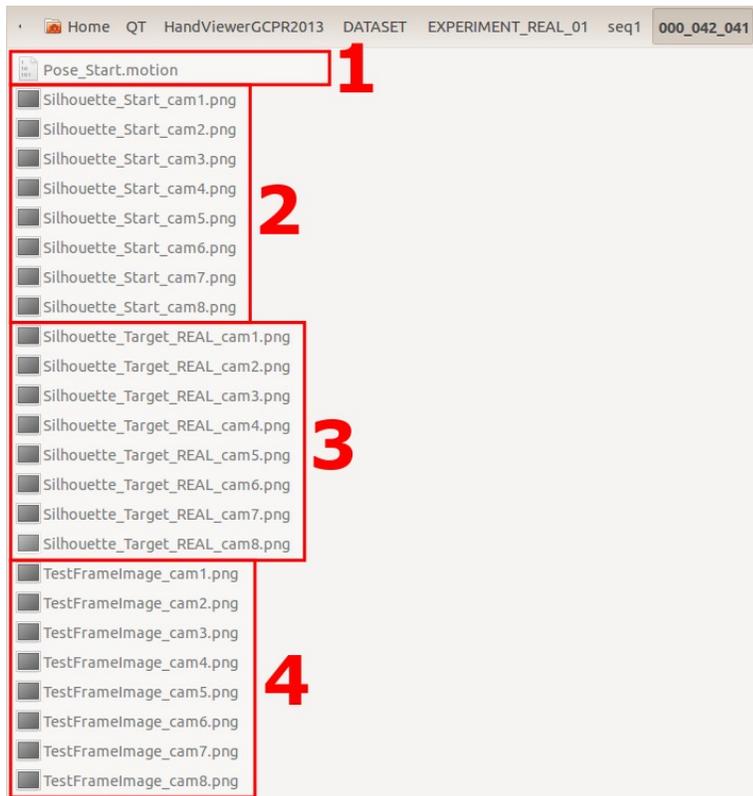


Each dataset folder, for both the realistic and synthetic part and for all frame differences contain 4 folders, as depicted above. These are the respective folders for all 4 sequences.

Each sequence folder contains sub-folders for its frame pairs. An example is depicted in the following picture. The contents of these folders, for both the realistic and the synthetic case, are described in the following sections.

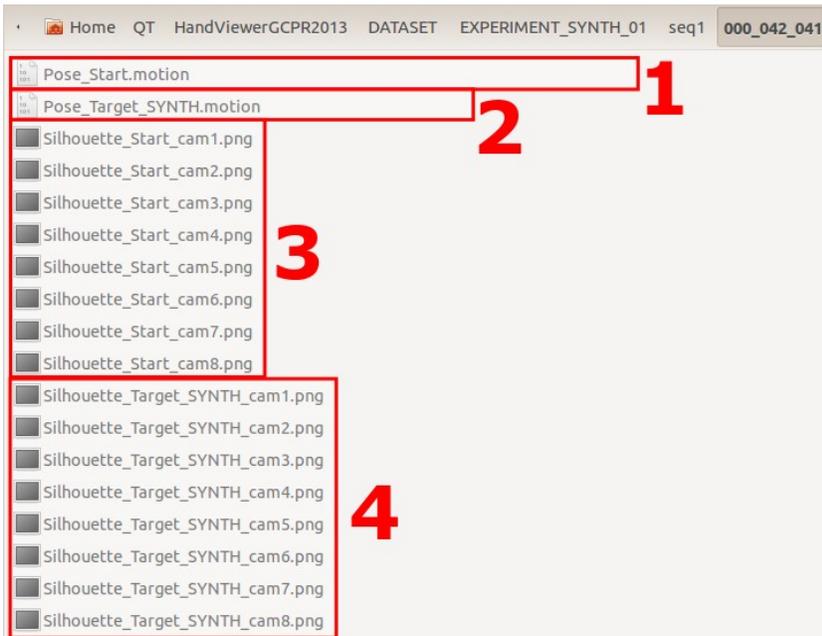


## Realistic part of Dataset



1. The Starting Pose. Example code on how to read this can be found in the file “mainwindow\_EXPERIMENT\_CLICK.cpp”, in the method “MainWindow::on\_myButton\_EXPERIMENT\_TestPose\_INI\_clicked()”.
2. Images depicting the Starting Silhouette for all camera views.
3. Images depicting the Target Silhouette for all camera views. (created from the actual camera-frames)
4. Images depicting the actual camera frame for all camera views.

## Synthetic part of Dataset



1. The Starting Pose. Example code on how to read this can be found in the file “mainwindow\_EXPERIMENT\_CLICK.cpp”, in the method “MainWindow::on\_myButton\_EXPERIMENT\_TestPose\_INI\_clicked()”.
2. The Target Pose. Example code on how to read this can be found in the file “mainwindow\_EXPERIMENT\_CLICK.cpp”, in the method “MainWindow::on\_myButton\_EXPERIMENT\_TestPose\_FIN\_clicked()”.
3. Images depicting the Starting Silhouette for all camera views.
4. Images depicting the Target Silhouette for all camera views. (created from the synthetic Target Pose)

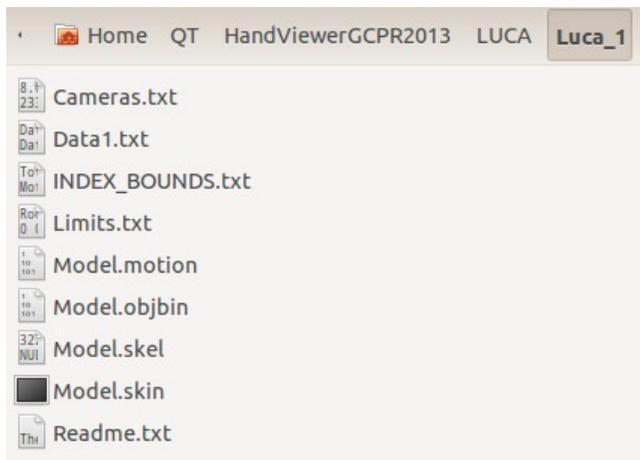
## Folder “LUCA”

This folder contains some of the files released by Ballan et al. (<http://cvg.ethz.ch/research/ih-mocap/>)



Four folders are included, each containing all the necessary files for each sequence

(Note that there's some redundancy, so that each sequence folder can be independent).



Each folder contains the same file structure, as depicted in the left image.

Description for all the files can be found in a text written by Luca Ballan - <http://cvg.ethz.ch/research/ih-mocap/FileFormat.txt>

Example code on how to read them can be found in the files/methods contained in the following table.

Description	File in Folder	File in Project	Method
Camera Matrices	Cameras.txt	cameraSet.cpp	CameraSet::setupCameras()
Mesh	Model.objbin	model_Mesh.cpp	Model::readMesh_Luca()
Skeleton/Kin.Chain	Model.skel	model_Skeleton.cpp	Model::readSkeleton_Luca()
Skin	Model.skin	model_Skin.cpp	Model::readSkin_Luca()
Motion Data	Model.motion	animation_Files_CompleteSequence.cpp	Animation::readFromBinary()
Limits/DoFs	Limits.txt	model_Limits.cpp	Model::readLimits_Luca()
Detailed Instructions	<a href="http://cvg.ethz.ch/research/ih-mocap/FileFormat.txt">http://cvg.ethz.ch/research/ih-mocap/FileFormat.txt</a> (written by Luca Ballan)		