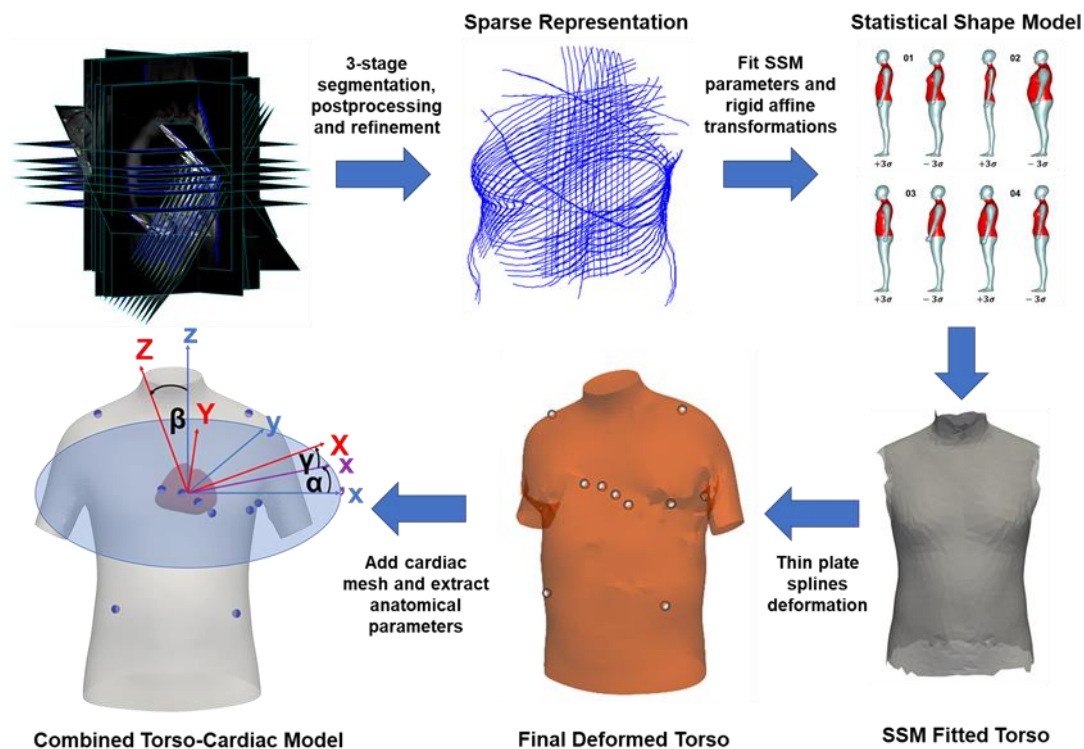


Torso Reconstruction Pipeline Documentation

Pipeline overview

This pipeline is designed to take cardiac magnetic resonance (CMR) images from the UK Biobank (Petersen, Matthews et al. 2013) and reconstruct the subject's torso surface and localise their ECG electrode locations. The workflow is illustrated below. The reference for the contour extraction is Smith et al. 2022 (Smith, Banerjee et al. 2022), and the full pipeline is currently in preprint (Smith, Rodriguez et al. 2023), but check if it is now out.



Briefly, it operates as follows:

- (1) Unpacks the correct CMR images from the DICOMs.
- (2) Extracts the refined torso contour outlines from the images. The full method is explained in Smith et al. 2022 (Smith, Banerjee et al. 2022), but briefly it segments into white torso, black background with a ML network (u-net (Ronneberger, Fischer et al. 2015)), then draws the largest single continuous contour around the torso, and then refines the resulting contour by removing the head, neck, arms, and artifact distortions with a second u-net. The resulting refined contours give a sparse representation of the torso.
- (3) Fits the first 40 components of a statistical shape model (Pishchulin, Wuhrer et al. 2017) of the human body to the contours.

- (4) Deforms the resulting torso towards the refined contours using a thin plate splines methodology (Villard, Grau et al. 2018).
- (5) The ECG electrodes in a Mason-Likar system are placed on standard positions on the mean statistical shape model and then transformed alongside the mesh to give personalised electrode positions.
- (6) This gives a torso with electrode positions that you can use as you like! E.g. combining with a personalised cardiac mesh (Banerjee, Camps et al. 2021, Beetz, Banerjee et al. 2023, Doste, Camps et al. 2025).

Why personalise the torso?

Clinical, experimental, and computational studies show that the relative position and orientation of the heart with respect to the torso affects the ECG, including durations, amplitudes, and composite measures (MacLeod, Lux et al. 1998, Nguyễn, Potse et al. 2015, Mincholé, Zacur et al. 2019, Smith, Rodriguez et al. 2023). This means that if you want to personalise an anatomy for digital twinning and *in silico* trials, the torso is an important part of that (Corral-Acero, Margara et al. 2020, Camps, Berg et al. 2024), alongside studying the cardiac structure-function relationship itself.

Step by step instructions

This install was performed on a desktop running windows 10 (processor: intel core i7-9700) and requires python 3.8.3 on the device

- (1) **Download TORSO-MPP** from <https://github.com/MultiMeDIA-Oxford/TORSO-MPP> - note that if your path to this contains spaces (e.g. as in OneDrive - Nexus365) this can cause issues – avoid doing this.
- (2) **Set up the required virtual environment** – this has a few dependencies on packages and can be fussy about the version compatibility. I will share exactly what versions I used, it's possible others work as well, but I will not guarantee it. I am using cuda 11.0 – others may not work (and at the very least you would have to change <https://download.pytorch.org/whl/cu110> to match your cuda distribution)
 - a. *Create and activate environment.* On the command line (while you're in the folder that you want the virtual environment to be – note this down as you'll need the path later) run:

```
py -3.8 -m venv "torso-2D-seg"
```

```
.\torso-2D-seg\Scripts\activate
```

(NOTE: feel free to change the name of this as long as you're consistent later when changing the paths)
 - b. *Install dependencies.* On the command line run, in this order:

```
py -m pip install --trusted-host pypi.org --trusted-host files.pythonhosted.org future
```

```
py -m pip install --trusted-host pypi.org --trusted-host files.pythonhosted.org dataclasses
```

```
py -m pip install --trusted-host pypi.org --trusted-host files.pythonhosted.org matplotlib
```

```
py -m pip install --trusted-host pypi.org --trusted-host pypi.python.org --trusted-host download.pytorch.org torch==1.7.0 torchvision==0.8.0 -i
```

```
https://download.pytorch.org/whl/cu110
```
- (3) **Change paths** where required in code
 - a. *MPP_Torso_Contouring_main*
Line 3: `run('C:\Users\hanith\OneDrive - Nexus365\TorsoReconstruction\Torso_Contouring_Hannah\MPP\mppSETUP.m');`
Change to the path of your mppSETUP.m file, which is in the main TORSO-MPP folder on github.
Line 4: `directf = 'C:\Users\hanith\OneDrive - Nexus365\TorsoReconstruction\Torso_Contouring_Hannah';`
Change to the path to the folder that you downloaded your TORSO-MPP folder into (the main one that everything's in).
Line 30: `mppOption pathfull = ['C:\Users\hanith\OneDrive - Nexus365\TorsoReconstruction\Torso_Contouring_Hannah\MPP\'];`
Change to the path to your TORSO-MPP folder.
Line 33: `cd(directf); files = dir(fullfile(directf, 'data'));`
Your subject data should be in folders in a data directory in the same place as the TORSO-MPP folder. E.g. if you saved TORSO-MPP in Torso-Reconstruction, your first subject should have a path Torso-Reconstruction\data*subject_no_1*. If you want to change the name data to something else, then change this line.

Line 36: `for ff = 1:length(files), SUBJECT_DIRS{ff} = fullfile(directf, 'data', files(ff).name); end`

Likewise for line 33.

- b. *mpp_Segmented_Contours_Torso* (saved in MeshPersonalizationPipeline)

Line 18: `setenv('PATH', ['C:\python-venv\torso-2D-seg\Scripts', pathsep, getenv('PATH')]);`

Change to the path of the Scripts folder in the location you created your virtual environment.

Line 19: `system(['"C:\python-venv\torso-2D-seg\Scripts\python" "', pathfull, 'PreTrained\torso_segmentor.py"', ' --dir_img ', Fullfile('mpp', 'torso-images'),...`

Change again to the path of python within the Scripts folder of your virtual environment. The torso_segmentor path does not need altering.

- c. *mpp_Automated_Contours_Torso* (saved in MeshPersonalizationPipeline)

Line 26: `setenv('PATH', ['C:\python-venv\torso-2D-seg\Scripts', pathsep, getenv('PATH')]);`

Change as previously to your virtual environment Scripts folder.

Line 27: `system(['"C:\python-venv\torso-2D-seg\Scripts\python" "', pathfull, 'PreTrained\torso_contouring.py"', ' --dir_img ', Fullfile('mpp', 'torso-images'),...`

Change as previously to your virtual environment python file.

Now your scripts are all ready to go!

- (4) **Upload subject data** – again I recommend a data folder in the same place as your TORSO-MPP folder. Each subject should be saved in a folder under data with a descriptive name (e.g. their subject ID) and their DICOMS folder from UKB within that. That's the DICOMS of the CMR imaging – so your path should e.g. look like:

Torso-Reconstruction\data*subject_no_1*\DICOMS\InlineVF*files* and so on for the different folders.

- (5) **Run the pipeline!**

- a. *Setup*: run **lines 1-36** (inclusive) of *MPP_Torso_Contouring_main* . This just sets paths and options.

NOTE: the remaining steps can easily be run over all subjects in the for loop shown, I will demonstrate running it for each subject

- b. *Set subject directory*: run **line 39** with your chosen file index (ff, e.g. 1 for your first subject).
- c. *Unpack DICOMS*: run **line 42** (*mpp_Read_DICOMS*). This compiles images, positions, and orientations into *mpp\DICOMS.mat* . This takes 1-2 minutes on my spec.
- d. *Extract subject data*: run **line 43** (*mpp_Subject_Data*). This compiles information on the subject into *mpp\SubjectData.txt* . This takes seconds on my spec.
- e. *Select heart slices*: run **line 44** (*mpp_Select_Heart_Slices_Biobank*). Selects the slices (*mpp\HS.mat*) and gives a montage showing their orientation (*Montage_slices.png*). This takes seconds on my spec.
- f. *Refine slices for torso*: run **line 45** (*mpp_Get_AllPosition_Images_Torso*). Prunes the slices and adds positional information to *mpp\BS.mat*). Takes seconds on my spec.
- g. *Segment the torso*: run **line 47** (*mpp_Segmented_Contours_Torso*). Takes the input CMR images (saved now in *mpp\torso-images*) and places a white mask over any body it sees using a u-net in your python environment, segmented images are saved in *mpp\torso-segment* and compiled in *mpp\BC0.mat* . This takes seconds on my spec.

TROUBLESHOOTING: did you change the paths to your python environment in the mpp_Segmented_Contours_Torso file? Does your path have spaces in (move to a directory that doesn't)? Have you got the correct versions of python, torch, and torchvision?

- h. *Extract and refine contours:* run **line 49** (mpp_Automated_Contours_Torso). Extracts the single largest continuous contour (saved in mpp\torso-contours-full) from the segmented images and then removes undesirable features using a second u-net (saved in mpp\torso-contouring) and compiles in BC1.mat. This takes seconds on my spec.

TROUBLESHOOTING: did you change the paths to your python environment in the mpp_Segmented_Contours_Torso file? Does your path have spaces in (move to a directory that doesn't)? Have you got the correct versions of python, torch, and torchvision? Did you install matplotlib?

- i. *Fix torso surface to contours:* run **line 51** (mpp_Fit_Vest_Biobank). Firstly fits the first 40 components of a statistical shape model of the human body (Pishchulin, Wuhrer et al. 2017) to the refined contours and then deforms this iteratively using a thin plate splines methodology (Villard, Grau et al. 2018). Outputs a full fitted body surface in mpp\BODY0.vtk and torso in mpp\VEST0.vtk . This takes about an hour on my spec.
- j. *Localise ECG electrode locations:* run **line 52** (mpp_ECG_Electrodes). Outputs the ten electrode positions in the standard Mason-Likar system widely used in ECG simulations. These have been manually placed on the mean SSM and then they are transformed alongside the body surface so they will be in standard positions on the fitted torso. Outputs in ECG_ELECTRODES.mat as a structure given as [x,y,z] position for each in millimetres. This takes seconds on my spec.
- k. *Refine torso:* run **line 53** (mpp_Final_Torso_Meshes_Biobank). Refines the fitted torso and outputs this as TORSO.vtk and ECG_ELECTRODES.vtk . This takes ~2 minutes on my spec.

Banerjee A, *et al.* (2021). "A completely automated pipeline for 3D reconstruction of human heart from 2D cine magnetic resonance slices." Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences **379**(2212): 20200257.

Beetz M, Banerjee A, Ossenberg-Engels J, and Grau V (2023). "Multi-class point cloud completion networks for 3D cardiac anatomy reconstruction from cine magnetic resonance images." Medical Image Analysis **90**: 102975.

Camps J, *et al.* (2024). "Digital twinning of the human ventricular activation sequence to Clinical 12-lead ECGs and magnetic resonance imaging using realistic Purkinje networks for in silico clinical trials." Medical Image Analysis **94**: 103108.

Corral-Acero J, *et al.* (2020). "The 'Digital Twin' to enable the vision of precision cardiology." European Heart Journal **41**(48): 4556-4564.

Doste R, *et al.* (2025). "An Automated Computational Pipeline for Generating Large-Scale Cohorts of Patient-Specific Ventricular Models in Electromechanical In Silico Trials." arXiv preprint arXiv:2503.03706.

MacLeod RS, Lux RL, and Taccardi B (1998). "A possible mechanism for electrocardiographically silent changes in cardiac repolarization." Journal of Electrocardiology **30**: 114-121.

Mincholé A, Zacur E, Ariga R, Grau V, and Rodriguez B (2019). "MRI-based computational torso/biventricular multiscale models to investigate the impact of anatomical variability on the ECG QRS complex." Frontiers in Physiology **10**.

Nguyễn UC, *et al.* (2015). "An in-silico analysis of the effect of heart position and orientation on the ECG morphology and vectorcardiogram parameters in patients with heart failure and intraventricular conduction defects." Journal of Electrocardiology **48**(4): 617-625.

Petersen SE, *et al.* (2013). "Imaging in population science: cardiovascular magnetic resonance in 100,000 participants of UK Biobank - rationale, challenges and approaches." Journal of Cardiovascular Magnetic Resonance **15**(1): 46.

Pishchulin L, Wuhler S, Helten T, Theobalt C, and Schiele B (2017). "Building statistical shape spaces for 3D human modeling." Pattern Recognition **67**: 276-286.

Ronneberger O, Fischer P, and Brox T (2015). "U-Net: Convolutional networks for biomedical image segmentation". Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, Cham, Springer International Publishing.

Smith HJ, Banerjee A, Choudhury RP, and Grau V (2022). "Automated torso contour extraction from clinical cardiac MR slices for 3D torso reconstruction". 2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC).

Smith HJ, *et al.* (2023). "Anatomical basis of sex differences in human post-myocardial infarction ECG phenotypes identified by novel automated torso-cardiac 3D reconstruction." arXiv preprint arXiv:2312.13976.

Villard B, Grau V, and Zacur E (2018) "Surface mesh reconstruction from cardiac MRI contours." Journal of Imaging **4** DOI: 10.3390/jimaging4010016.