Users Manual

ver. 1.0.0

Tommy Clausner - Sarang S. Dalal - Maité Crespo-García

Contents

1 Introduction and requirements

janus3D is an open source MATLAB toolbox for the purpose of EEG electrode determination and co-registration of 3D head models with individual structural MR images. It features smart selection and automatic texture-based EEG electrode detection, providing highly accurate EEG sensor positions for source reconstruction analyses. It makes use of GUI-based functions allowing the user to obtain sensor positions fast and comfortable. To enhance the actual 3D reconstruction process a tool for automatic background selection and mask creation is included. Electrode cap templates of EEG cap systems that are not included can be created using the implemented 'Template Builder'. This tool allows the creation of individual cap templates that are used for automatic labeling of the EEG electrode positions. janus3D requires MATLAB 2015a (MathWorks), including Image Processing Toolbox (MathWorks), Computer Vision System Toolbox (MathWorks), and Fieltrip Toolbox (Oostenveld, Fries, Maris, & Schoffelen, 2011). Due to highly working memory-consuming operations, at least 4GB of RAM should be dedicated to janus3D or MATLAB. A graphics card employing OpenGL is advantageous, to make all graphical operations as fluent and renderings as detailed as possible. Note that all performed 3D operations are done on high-resolution 3D models with an amount of hundreds of thousands vertex points. Most variables within janus3D are stored as global variables, meaning that they remain in the system's working memory until MATLAB is closed or they are deleted manually. Global variables can be accessed from the base workspace of MATLAB, but must not necessarily appear there. janus3D will remove all global variables at each start and also if it is terminated via the 'done & exit' button or deletion of the GUI window. If janus3D crashes global variables might remain stored in working memory. Deletion of global variables is possible via the MATLAB commands 'clearvars -global' or 'clearvars -global variablename'. They can be accessed via the 'global variablename' command.

2 Getting started

To get started with janus3D it is necessary to add the main folder of janus3D to the MATLAB search path. After typing 'janus', 'janus3D' or 'j3d' into the MATLAB command line window and pressing RETURN, the GUI window will show up. An illustration of the main window after starting janus3D is depicted in Figure 1. The main window consists of 4 different sections: the operation window, the main functions, the toolbar and the main function dependent sub-functions. Within the operation window, which

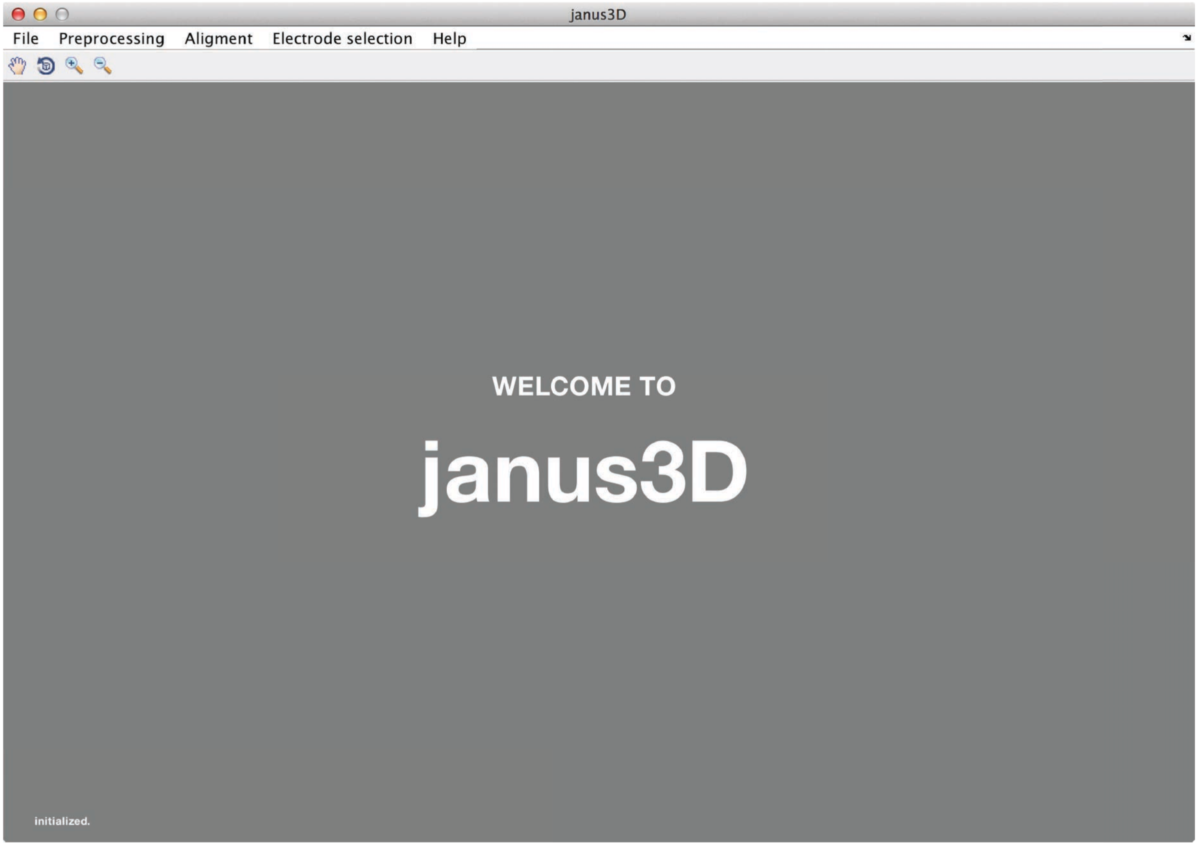

Figure 1: **janus3D main window**

is the large field in the center of the janus3D window, all visual outputs will be displayed. At the beginning of each session it displays 'Welcome to janus3D'. Its output will be adapted to current processing stages and displays all output graphics of ongoing operations. Therefore the operation window is the playground where all manipulations are done. On the topside of the main window, the main functions are placed. Those are designed as drop down menus giving access to all main functions janus3D provides. A tree diagram for all janus3D main functions is shown in Figure 2. To enhance usability and to ensure that functions are called in their respective order, all functions that miss relevant input data are grayed out and cannot be used at the respective stage. Selecting [File] expands all session relevant features like saving or loading of janus3D sessions, as well as giving access to the 'Photo Masker' and 'Template Builder' tools and the preferences section, where the default EEG cap template is set. [Preprocessing] includes all relevant steps for preparing the MRI mesh and the reconstructed 3D model for later steps. Clicking the [Alignment] button will expand functions used to co-register the

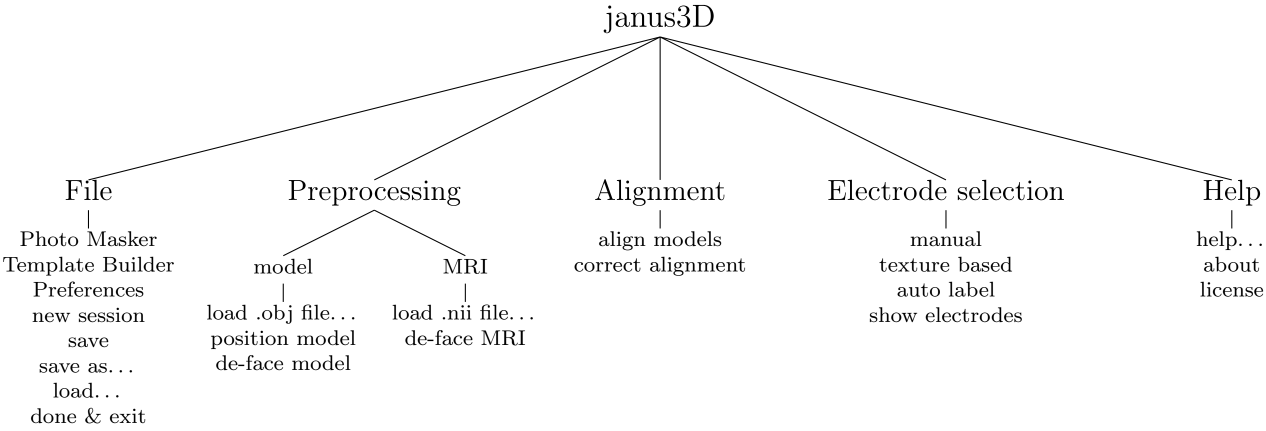

Figure 2: **janus3D main functions**

model to the respective structural MRI. Inside [Electrode selection] functions for manual and texture based electrode selection, automatic labeling and displaying the selected electrodes are included. The toolbar is located in the top left corner of the janus3D window, below the main functions section and is used to perform basic operations on the operation window. Those can be pan, rotate and zoom in the same way as known from common MATLAB figure windows. Some main functions provide different sets of sub-functions. All main function related sub-functions are displayed once the related main function was called. They remain invisible the rest of the time. Sub-functions are displayed on the right hand side of the janus3D window. Clicking the load button within the [Preprocessing] section imports the desired file. Depending on computer's specifications and the resolution of the input models, this can take up to several minutes. 3D objects can only be imported in Wavefront Object File format (*.obj) and can contain vertex coordinates, vertex texture coordinates and face indices. The import of vertex normals is not supported. Structural MR images are imported in NIfTI file format (*.nii). janus3D instantaneously segments the MRI and obtains its outer surface (scalp), which is used to create the actual MRI 3D model. See Figure 3 for an example on how to load MRI files. This function is based on the 'ft_read_MRI', 'ft_volumesegment', and 'ft_prepare_mesh' functions implemented in Fieldtrip (Oostenveld et al., 2011). Furthermore Table 1 shows a schematic depiction of a final janus3D session file.

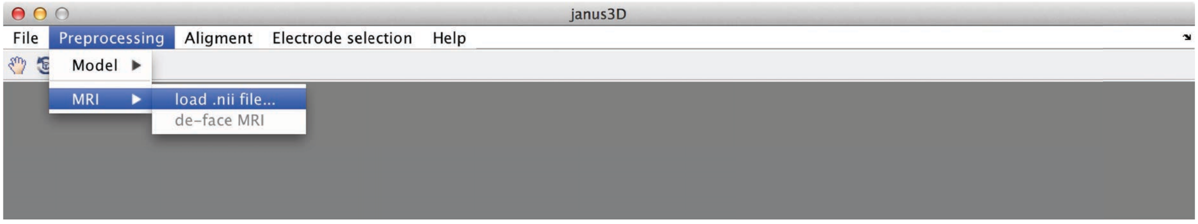

Figure 3: **janus3D load files**

Table 1: janus3D example structure of a session files

Structure Field	Annotation
model:	information related to the model (cf. Table 3)
MRI:	information related to the MRI
MRI.pnt:	original vertex coordinates as obtained by fieldtrip
MRI.tri:	original indices building a triangular face as obtained by fieldtrip
MRI.unit:	Unit representing the scale of vertex coordinates as obtained by fieldtrip
MRI.cfg:	Configuration file as used by fieldtrip
MRI.VCoord:	Vertex coordinates used by janus3D
MRI.FCoord:	Column 1,3,5 vertex indices building a triangular face as used by janus3D
MRI.Cut_Model:	Vertex and face coordinates of the face selection polygon
MRI.Face:	Vertex and face coordinates of the selected face
MRI.nonFace:	Vertex and face coordinates of all non-facial parts
Log:	information about all processing steps

3 Mesh preparation

Since the 3D meshes exported from PhotoScan are arranged in an arbitrary coordinate system and arbitrary orientation, the first step will be the reorientation of those into MRI space. After this step the x-axis should point from the left ear to the right, y-axis from the back to the front and z-axis from the bottom to the top. This needs to be performed manually, by selecting [Preprocessing] → [Model] → [position model]. As shown in Figure 4 the actual rotation takes place from 3 different view angles, that can be set via the respective drop-down menu. Rotational steps can be set in the field 'set angle', by filling the respective angle in degree. Pressing RETURN will rotate the model relative to the perspective that was selected, according to the mathematical direction of rotation. This means positive view angles will cause janus3D to rotate the model counterclockwise. The first step will be to select the left side view and make the model upright relative to the horizontal line. Once this was done the tip of the nose and the inion point should be aligned horizontally. After that, the topside view will be selected. As a result of this rotational step, the tip of the nose should point to the top of the

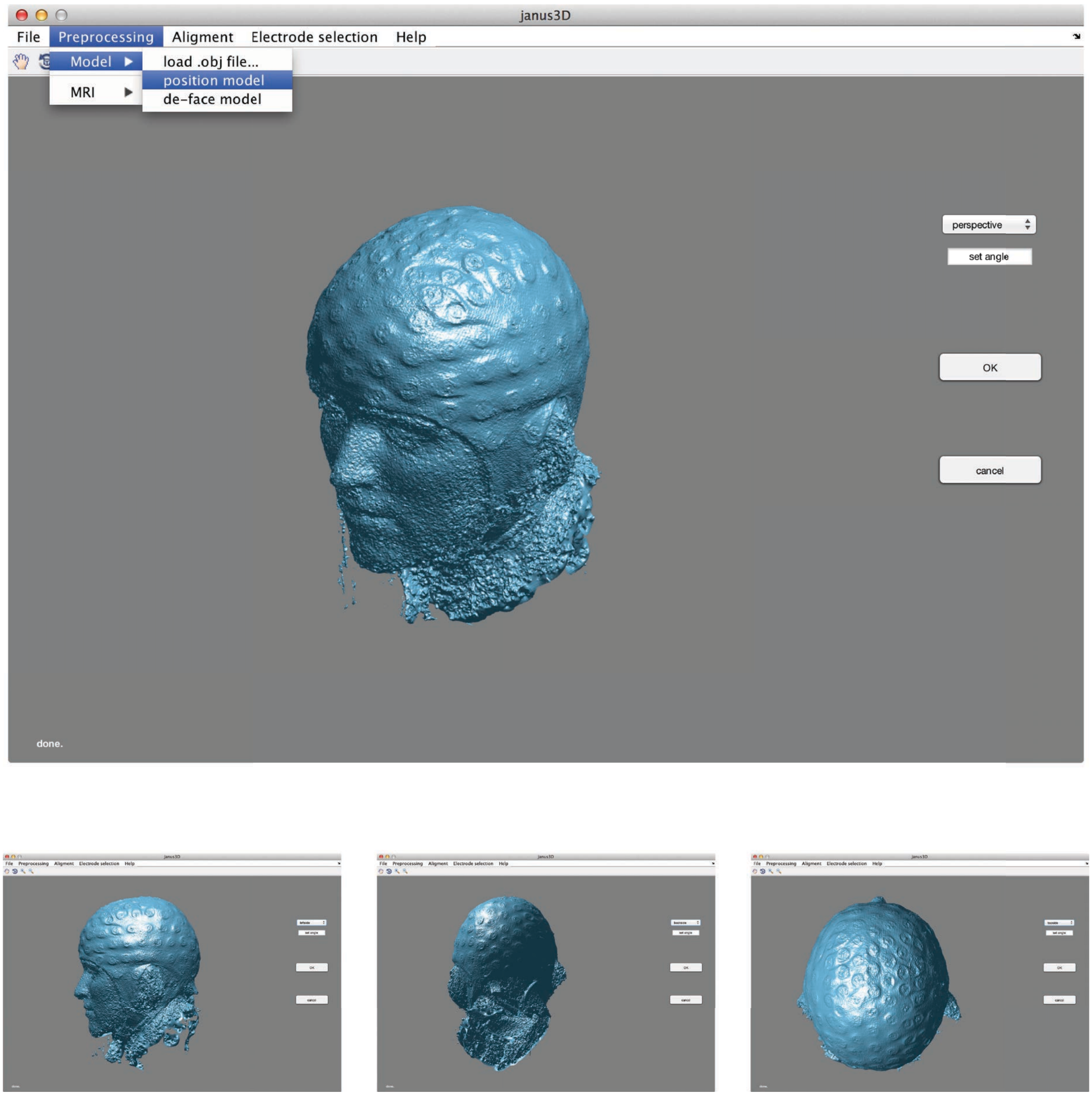

Figure 4: **janus3D mesh preparation**

Example result of the model's rotation into MRI space. The large scale image depicts the perspective view and the 3 small scale image exemplify how the model should look for the 3 respective view angles after a successful rotation. Using the 'set angle' field will rotate the model from its respective point of view according to the mathematical direction of rotation.

screen and the ears should be aligned on the horizontal line. The last step is to select the backside option in the drop-down menu and to rotate the model until the ears are horizontally aligned. To check whether the orientation was successful, it is advisable to go back to the perspective view and check, if the model looks to the bottom-left direction of the screen. This should be the exact same direction in which the MRI mesh 'looks' after importing a not manipulated NIfTI file. Clicking the 'OK' button applies the respective transformation to the model.

4 Face selection

The face selection is one of the core parts of janus3D. Selected facial parts from the 3D model and the respective MRI mesh are later used to align both models. Since this step requires a model pre-orientation into MRI space, because of the viewpoint specificity of the face selection, it will only be available after the model was oriented. The respective face selection procedure is located in [Preprocessing] → [Model/MRI] → [de-facing]. De-facing in that sense means that the face of each model will be selected (in terms of removed) for later iterative closest point matching. The easier way to go is to de-face the 3D model first and do the de-facing of the MRI mesh afterwards. Calling the de-facing functions will make specific main function dependent sub-functions available. Doing de-facing for the first model is possible only using the 'select by shape' function. Clicking this function will enable a point-and-click selection of facial parts, by framing them using sparse points. This step was depicted in Figure 5 (top left). Right clicking within the figure window will remove the previously selected point. The actual selection will be confirmed after clicking 'select by shape' again. By selecting 'perspective' from the drop-down menu it is possible to inspect the model from a perspective view indicating the current selection, as shown in Figure 5 (top right). However the selection itself can only take place in side view. Due to the fact, that the upper part of the face includes more morphological invariant parts, a facial selection from the eyebrow to the upper lip can be more sufficient than selecting the face as whole. As long as either the model or the MRI was not 'de-faced', the 'select by shape' option remains grayed and cannot be used. This is because the 'select by shape' option is based on the boundary of the previously framed shape. Once either the model or the MRI was de-faced, the respective second one can be selected by clicking 'select by shape', which now will be an available option. After clicking the button the mouse cursor changes imitating the boundary shape of the previously selected face. After fitting the shape to the respective model, clicking the left mouse button indicates the selection. Due to the arbitrary scaling of the model exported from PhotoScan, the boundary shape might slightly vary in size and rotation. By using the mouse wheel it is possible to freely rotate the shape and by pressing and holding the CTRL key, while using the mouse wheel, the scale can be adapted. Using those features it is possible to fit the boundary shape to the silhouette of the second mesh. In both cases the final selection will only be applied after clicking the 'OK' button. An example selection is shown in Figure 5 (bottom left and right).

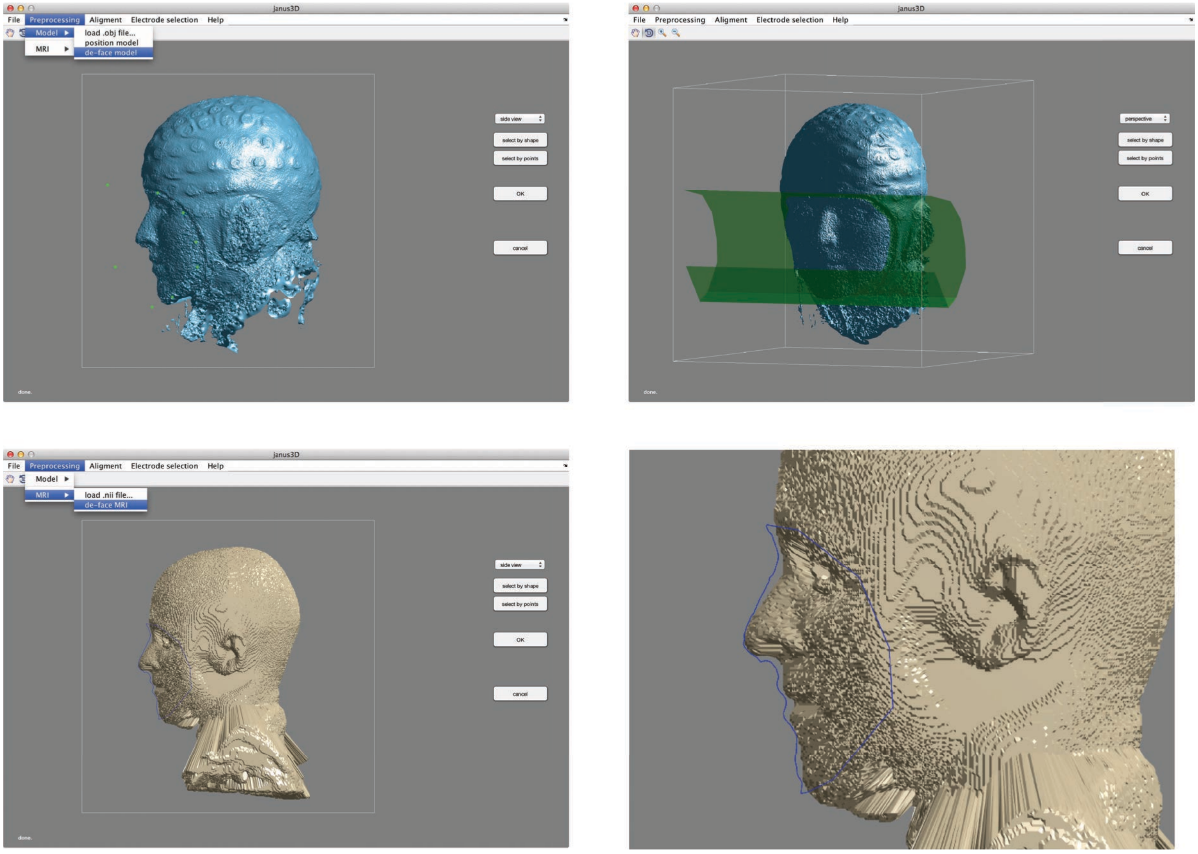

Figure 5: **janus3D face selection**

Exemplified selection by using the 'select by points' method (top) and the 'select by shape' method (bottom). The latter will be available after the first face selection took place.

5 MRI co-registration

Once both models are 'de-faced', janus3D will be enabling the 'align models' option located in [Alignment]. This function is based on an iterative closest point (ICP) algorithm, which in turn is based on the work of Chen and Medioni (1991) and Besl and McKay (1992). Prior to this step janus3D performs a center coordinate matching and scaling of both meshes bounding boxes. Before the actual co-registration takes place two options including two alternative settings have to be selected. One of those choices refers to the actual alignment that will take place. Either both meshes can be aligned 'Point to Point' or 'Point to Plane'. According to the MATLAB documentation the

Point to Plane algorithm improves the performance for planar surfaces (MathWorks, n.d.-d). This is recommendable if the selected facial parts are relatively flat. The second setup options refer to the method that is used to scale the polygon-mesh of the face of the model to the same scale as the face selection of the MRI. As there are 'scale by max' and 'scale by mean'. 'Scale by mean' computes the scaling based on the average distance of all vertex points to their respective mean value over all vertex points, relative to the same value obtained from the MRI's facial vertex coordinates. Whereas scale by max will scale the mesh according to the average of the maximum difference between vertex coordinates in each orthogonal direction, relative to those obtained from the MRI. Therefore 'scale by max' is useful if facial selections slightly differ, affecting the actual average value. Per default the options are set to 'scale by mean' and 'Point to Point'

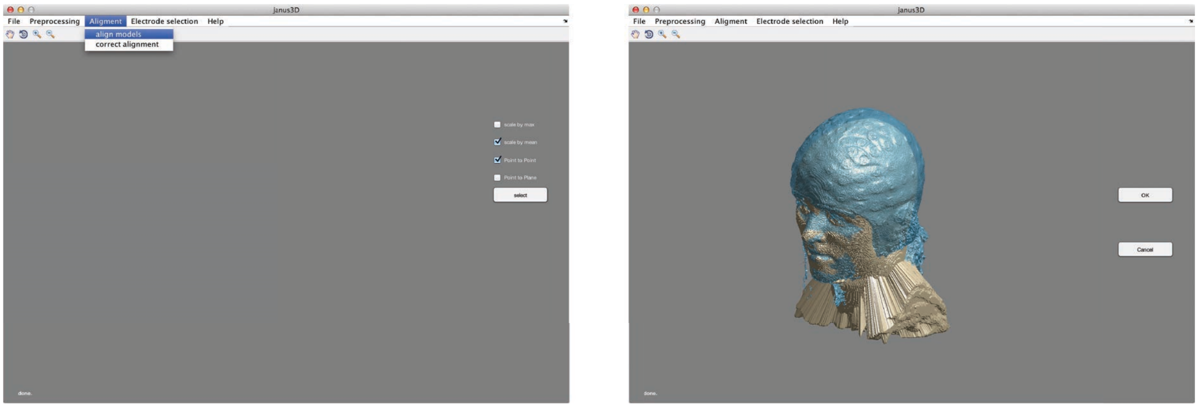

Figure 6: **janus3D align models**

as shown in Figure 6 (left), which will work for most alignments. After computing the ICP based alignment the transformation matrix used to transform the model's face to the face of the MRI, it is applied to the model as whole. The result of this step will be shown in the operation window as depicted in Figure 6 (right). In case the alignment was not satisficing it can be canceled by clicking 'cancel'. Adjusting the facial selection or removing mesh extensions that might have influenced the actual alignment, can help to overcome possible problems. If a proper alignment was not possible, janus3D offers to do a manual alignment. This can be done using the 'correct alignment' function, which will be enabled after the first attempt to automatically align the meshes. It offers seven different editable text boxes to correct translation, rotation and scaling respectively. The upper row of the text box array is used to translate the model in each of the three dimensions. The second row will rotate the model around each of the three axes of the coordinate system. The last row, which only consists of the field 'scale', allows scaling relative to 1. So typing '2' into the editable text box would cause janus3D to double the model's size. This step is shown in Figure 7.

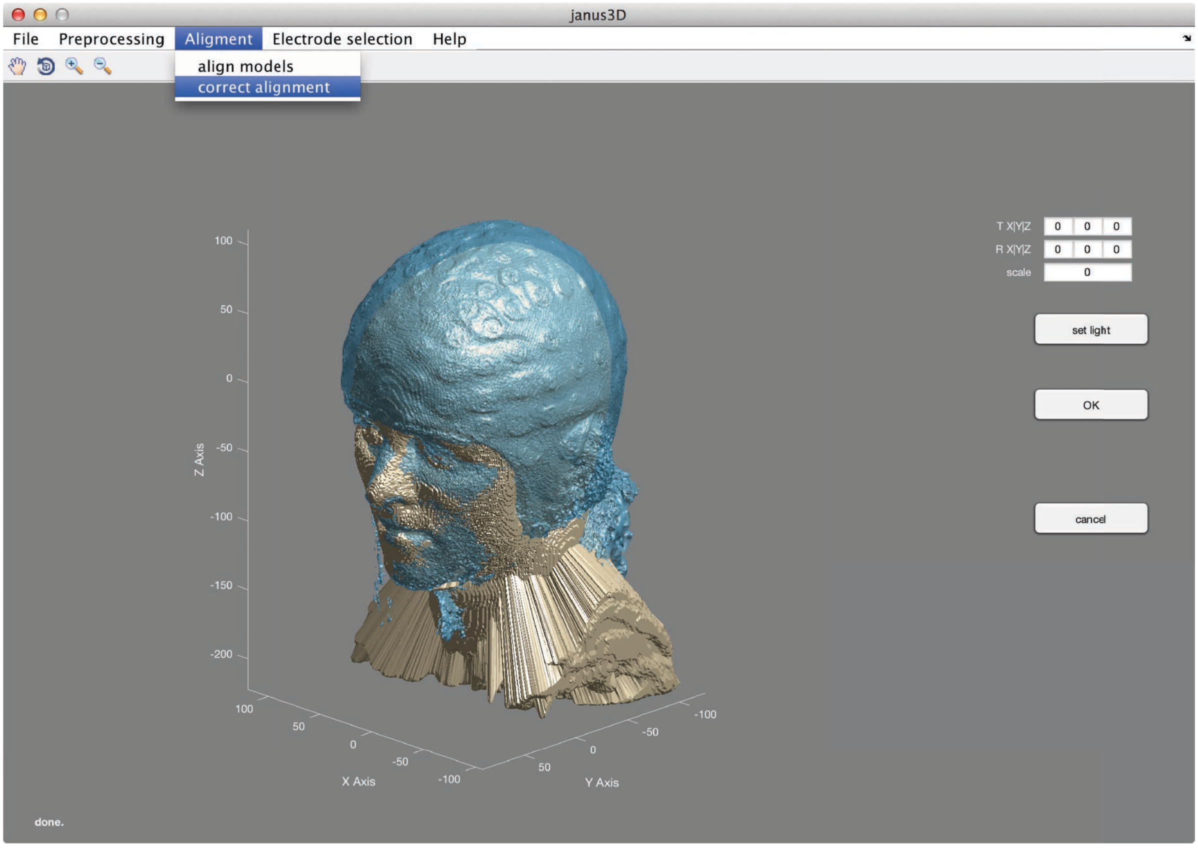

Figure 7: **janus3D correct alignment**

6 Electrode detection

Electrodes positions can either be determined manually or automatically based on the texture. Manual selection can be performed by selecting [Electrode selection] → [manual]. If a previously selected texture file was stored within the current session, janus3D will automatically apply this texture to the mesh. In case a texture file was not attached to the current session, two buttons will be present in the sub-functions section that otherwise will be not. The button 'select texture' opens a new dialog where the related texture file can be chosen. Notice that only JPEG texture files are supported so far. Manual electrode selection can either be done on the textured or un-textured mesh. Clicking the 'Set light' button will reset the current mesh light to the current view angle. This can be useful when electrodes are selected that are in the realm of shades.

Furthermore manual selection can either be performed 'free' or 'guided'. A free selection in that context means that any vertex point of the model can be selected and named by typing any text into the consecutive text box. In particular, this options can be used to set fiducial points or any points, that are not included in the 'guided' selection. Because defining a desired name for each selected point within the free selection would be a long lasting procedure, the guided selection offers the option to select electrodes in a predefined way. If an EEG cap template was defined within the preferences section, those labels are used in the respective order of the template file. janus3D will ask for a specific electrode to be set at next and labels it accordingly. The respective label will be displayed below the 'guided / free' section. Clicking 'guided' without having a template file set beforehand will cause janus3D to display a dialog where the desired file that should be used as a 'guide', should be selected. This guidance file can either be a *.txt or a *.mat file (n x 1 cell array), including the respective electrode labels. Points determined by guided selection will be stored in a different way than points obtained by the free selection. The latter one cannot be used for being projected onto the MRI's surface. Once the actual electrode selection is finished pressing either 'add' will add the current selection to previously selected electrodes or 'replace' will replace the previous selection. Clicking 'merge' will merge free and guided selection so that both point clouds are stored as they were selected by guided selection only and therefore can be projected onto the MRI's surface. The electrodes can directly be exported to a *.mat file by clicking 'export'. Removing all electrodes from the current session file is possible using the 'clear all' function. A more comfortable way of selecting electrode positions is to use the build in automatic electrode detection, by choosing [Electrode selection] → [texture based] from the main functions menu. This function selects electrode positions based on the texture information provided by the model. If no texture file was selected in a previous step, janus3D will ask to select a texture file by showing the respective file selection dialog. Before the actual selection can take place some options have to be defined. The first step is to adjust the contrast of the exemplified image so that the actual electrodes are well described. Using the slide controller depicted in Figure 9 (top left) will adjust the threshold of the contrast based color dissociation. In some cases (e.g. if the actual electrodes are darker than the cap) it is necessary to use the 'invert' option, which inverts the actual image color. This is inevitable, since the implemented electrode detection algorithm assumes the electrodes to appear in a white manner. Once the contrast definition was finished a click to the 'Accept' button finally predefines the contrast value. Afterwards a 'typical' electrode needs to be defined, by dragging and

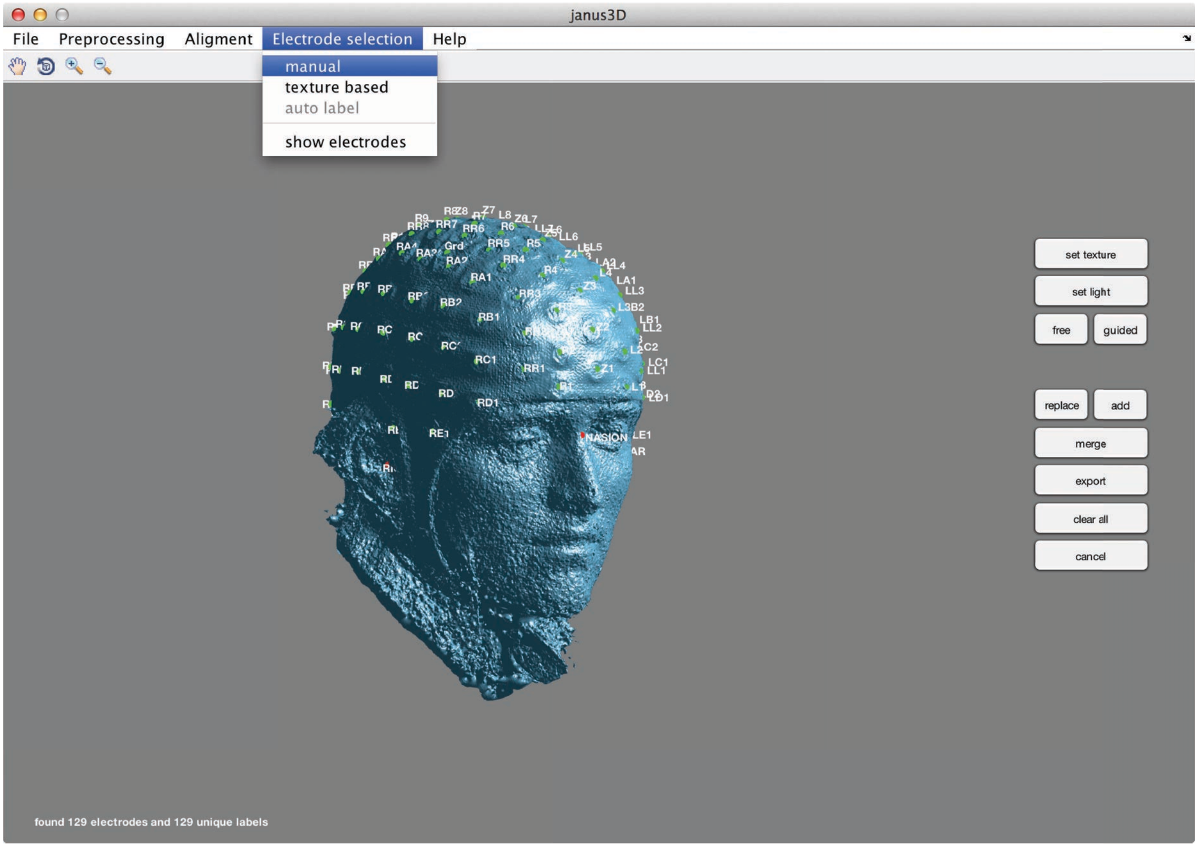

Figure 8: **janus3D manual electrode selection**

Showing the un-textured model with the corresponding selected set of electrodes (green) and fiducial markers (red). Green electrode markers indicate, that those can be projected orthogonally to the MRI surface. Red fiducial markers were selected using the 'free' selection and will not be projected onto the MRI surface. Note that once a texture file was attached to the janus3D session file, the model will be shown in textured fashion.

resizing the blue square so that it frames a single electrode. This step was depicted in Figure 9 (top right). The respective electrode will be selected after double clicking the rectangle and the automatic detection starts, showing the results in the operation window. As shown by Figure 9 (bottom) the textured mesh will appear depicting the centers of the electrodes marked in blue. Furthermore janus3D will output the current electrode count. If the number of detected electrodes or some positions does not match the expected values or locations, a manual correction can take place. By clicking 'correct' the manual correction function will be called. A left click to the meshes surface will add an electrode at this specific point in 3D space, whereas right clicking to any will remove it.

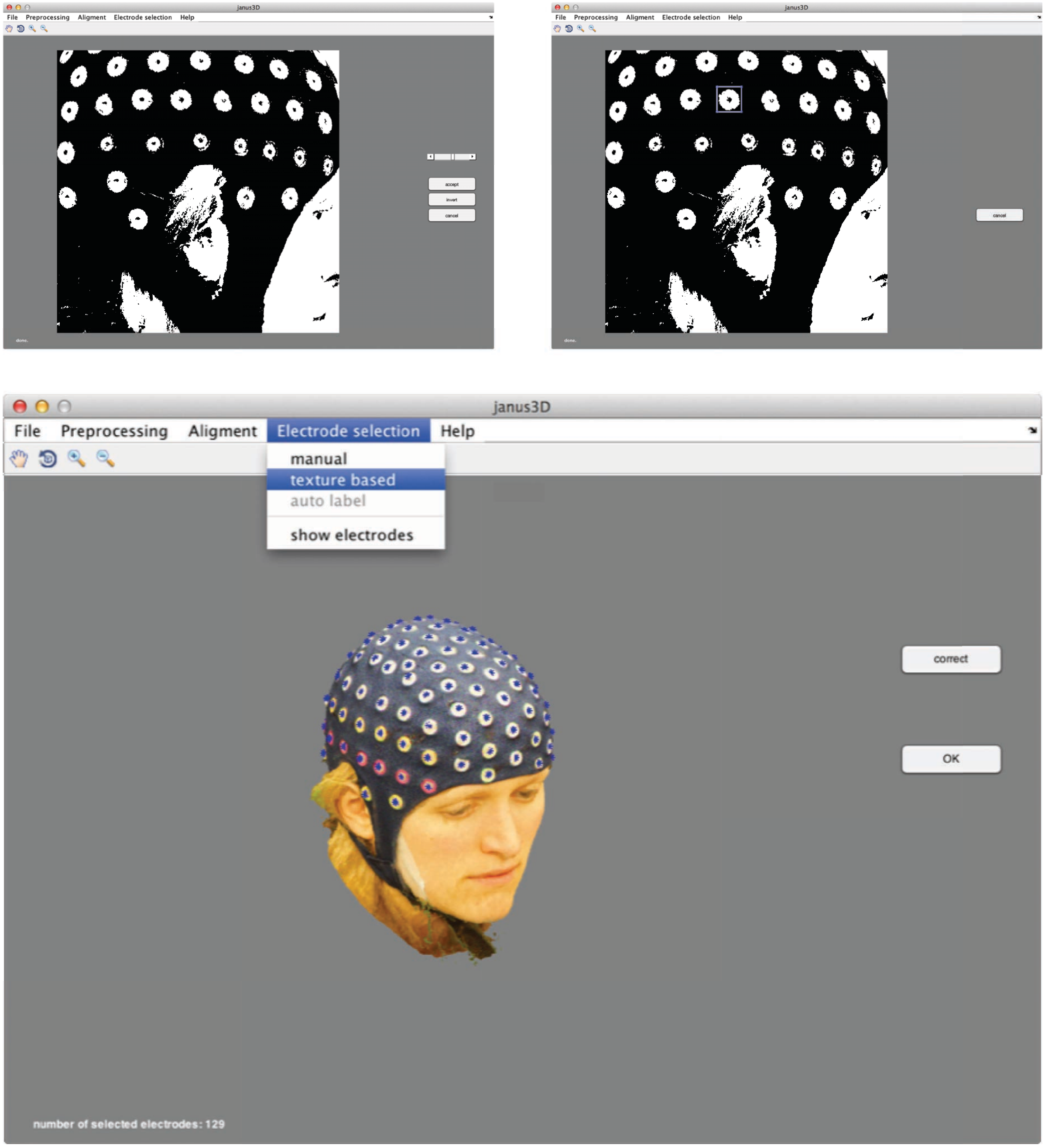

Figure 9: **janus3D texture based electrode selection**

Showing how contrast values should be adapted (top left), a template electrode is selected (top right) and the final textured representation of the textured mesh including all electrode positions indicated in blue (bottom).

7 Electrode labeling

Automatic electrode labeling is used to label electrode positions detected using the automatic electrode detection. This function is based on the template file pre-defined in the preference section. The algorithm finds the most plausible solution to fit the selected points to the template points and labels them accordingly. Afterwards the results are shown as a color-coded map of labeled points, which allows easy inspection and detection of outliers. Wrong-labeled electrodes can be corrected, by selecting them via left click, causing a drop-down menu to appear. There the actual correct label can be selected and will also cause the color map to instantaneously adopt the correct color. Once the selection is finished it is possible to directly export the current electrode set, by clicking 'export'. Notice, that for auto labeling the number of electrodes that are going to be labeled and the electrodes within the template file must match. Figure 10 depicts an example of a fully labeled ANT Waveguard 128 Channel electrode cap system. For demonstrational purpose, electrode L6 was selected. An overview about the actual electrode structure as created by janu3D is shown in Table 2.

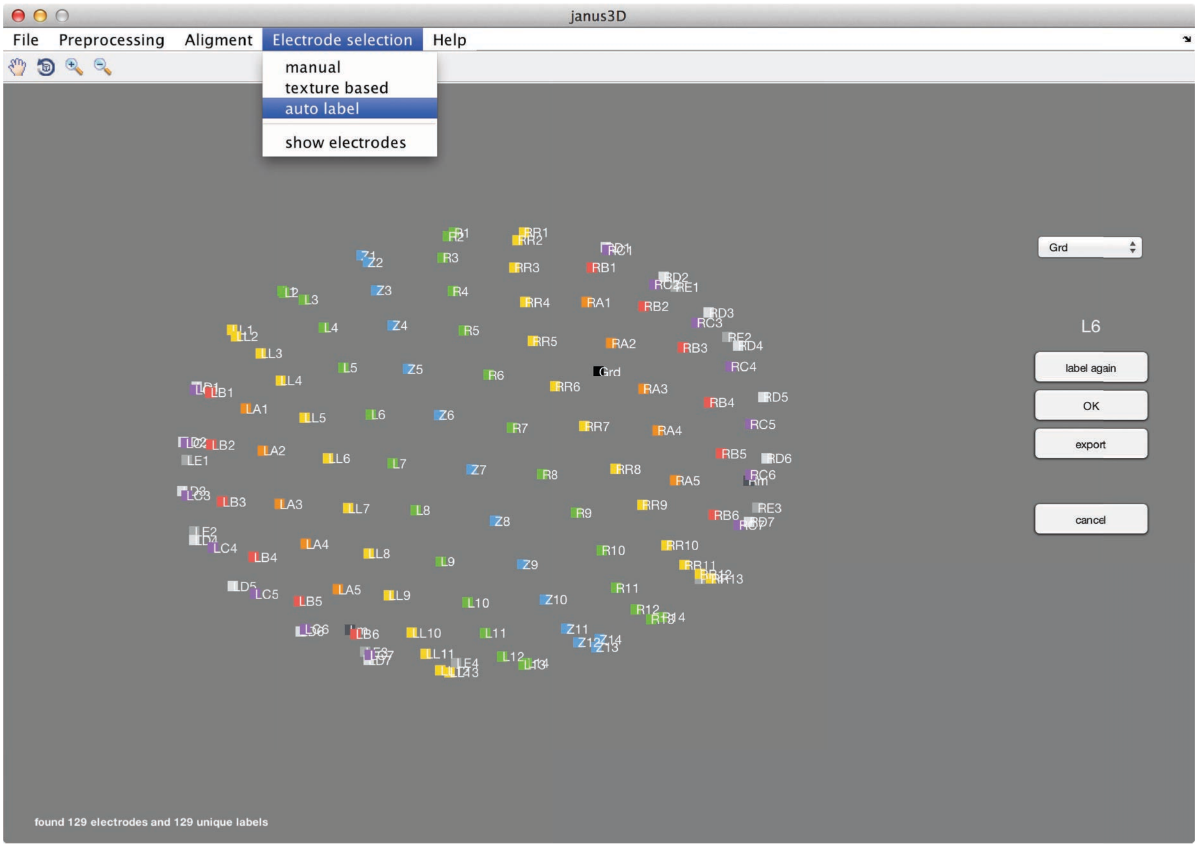

Figure 10: **janus3D automatic labeling**

Showing an exemplified result of janus3D automatic labeling. Electrodes were obtained using the texture based electrode selection (cf. Figure 9). Electrode L6 was selected for demonstration purpose.

Table 2: janus3D example output structure of exported electrodes

Structure Field	Annotation
model:	information related to the model
MRI:	information related to the MRI
model.points:	coordinates of electrode positions obtained from the model's surface
model.label:	labels of coordinates obtained from the model's surface
model.free_selection:	information related to the free selection
model.free_selection.points:	coordinates of points obtained by free selection
model.free_selection.label:	labels of coordinates obtained by free selection
MRI.points:	coordinates of electrode positions projected to the MRI's surface
MRI.label:	labels of coordinates projected to the MRI's surface

8 Plotting and output files

By pressing the 'show electrodes' button, the actual electrode positions are plotted on the 3D model and the MRI's surface. If the respective texture file was attached, the model's mesh will be shown in textured fashion. From this section it is possible to directly export the electrode by clicking the 'export' button. An example result is shown in Figure 11. When clicking the 'done & exit' button accessible from [File], the current processing state will be assigned to the base workspace of MATLAB. It contains a structure that is exemplified in Table 3.

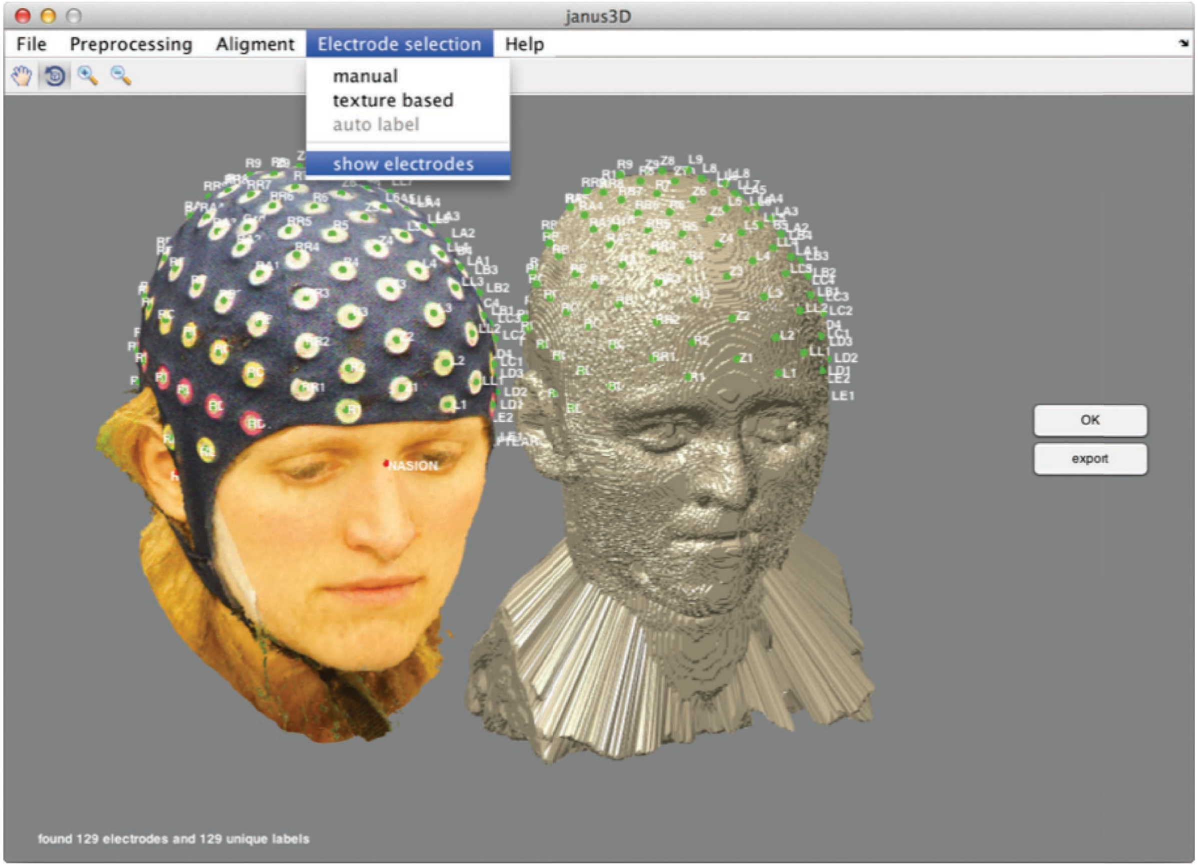

Figure 11: **janus3D example result**

Showing an exemplified result of a final electrode selection. After exporting them the electrodes file will contain the fields described in Table 2. Electrodes were obtained using the texture based electrode selection (cf. Figure 9) and automatic labeling (cf. Figure 10). Note that green electrode markers were orthogonally projected to the MRI surface, whereas red fiducial markers were not.

Table 3: janus3D example output structure of the final model

Structure Field	Annotation
VCoord:	Vertex coordinates
VIndices:	Indices of vertex coordinates from the original .obj file
VTCoord:	Texture coordinates
VTIndices:	Indices of texture coordinates from the original .obj file
FCoord:	Column 1,3,5 vertex indices building a triangular face
	Column 2,4,6 texture indices for triangular faces
FIndices:	Indices of faces from the original .obj file
ModelSpecs:	Additional information from the original .obj file
ModelSpecsIndices:	Indices of additional information from the original .obj file
transmat:	Transformation matrix
positioned:	Showing that the model was prepositioned
Cut_Model:	Vertex and face coordinates of the face selection polygon
Face:	Vertex and face coordinates of the selected face
nonFace:	Vertex and face coordinates of all non-facial parts
aligned:	Showing that the model was co-registered to the MRI
texture:	Attached texture file
Electrodes:	Electrodes determined on the model
Electrodes_on_MRI:	Electrodes projected to the MRI's outer surface (scalp)

9 Template Builder

The 'Template Builder' is a tool within janus3D to interactively create and manipulate templates of not included EEG electrode caps for automatic labeling. Opening janus3D and selecting [File] → [Template Builder] will give access to the function. It has its own GUI window, which is arranged in 4 different sections. The operation window is the large field in the center of the 'Template Builder' window. It will show the currently selected electrodes viewpoint dependent. Located in the upper right part of the main window, a D-pad for viewpoint specification can be found, to set the desired visual angle. Except for 'top', all electrodes that could not be seen physically are hidden. All main functions that 'Template Builder' provides are located at the topside. A schematic depiction of which functions are included can be found in Figure 12. When creating a color scheme for a template, it is useful to see a live preview of the actual state. For that reason a colored rotatable scheme will be displayed in the right lower corner of the window. Clicking next to it will create a larger preview window that allows basic MATLAB image manipulation, as shown in Figure 13 (bottom left).

To create a new template file at least one electrode file from what the template will be created is required. Because the auto labeling function of janus3D uses all sets

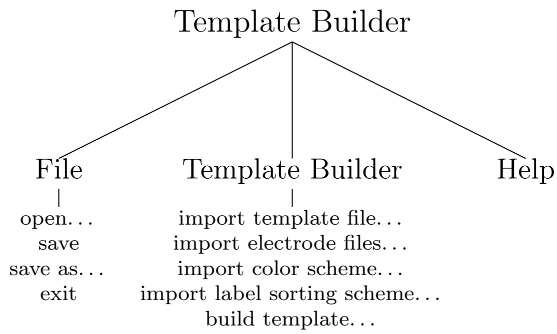

Figure 12: **Template Builder main functions**

of electrodes stored within the template file for making a label decision it should be considered using 5-10 sets of labeled electrodes to create a template. Using only one set would make the auto labeling inaccurate and using too many, would make it slow. By clicking [Template Builder] → [import electrodes files...] a dialog will show up, in which multiple files can be selected. Afterwards the first of the selected will be displayed in the operation window, including the respective electrode labels. The color will be gray for all electrodes, by default (cf. Figure 13 top left). Using the D-pad does the selection of the preferred viewpoint. Beside the actual file name used for storing the template file an editable text box asking 'set new template name' offers the option to specify the type of the template (e.g. 'ANT Waveguard 128 small'). Once selected, this name will be displayed inside the preference window of janus3D. For using a respective template file together with the auto-labeling function of janus3D, it is not necessary to specify a specific color schema. However a well-chosen coloring makes it easier to detect inaccuracy due to the auto label algorithm later. For some EEG cap systems stripes might be a good solution, but for others it might an arrangement of colored fields that make mislocalizations easier to be detected. Color codes can be assigned to the respective electrodes by clicking the white space between the electrodes. The cursor will change, now showing a small cross. Using this to draw an outline for all electrodes that shall go into will make the actual selection (cf. Figure 13 top right). Notice that once the mouse button will be released, the drawn line will automatically be closed. After double clicking the area enclosed by the blue line, a color selection window will appear, asking for the desired color choice. Electrodes that were already colored will be removed, from the operation window, to make further selections easier. The template file will be build, using the [Template Builder] → [build template...] function (cf. Figure 13 bottom right). Now the template file can be selected within the janus3D preferences and will be used for automatic electrode labeling. The 'Template Builder' also provides a

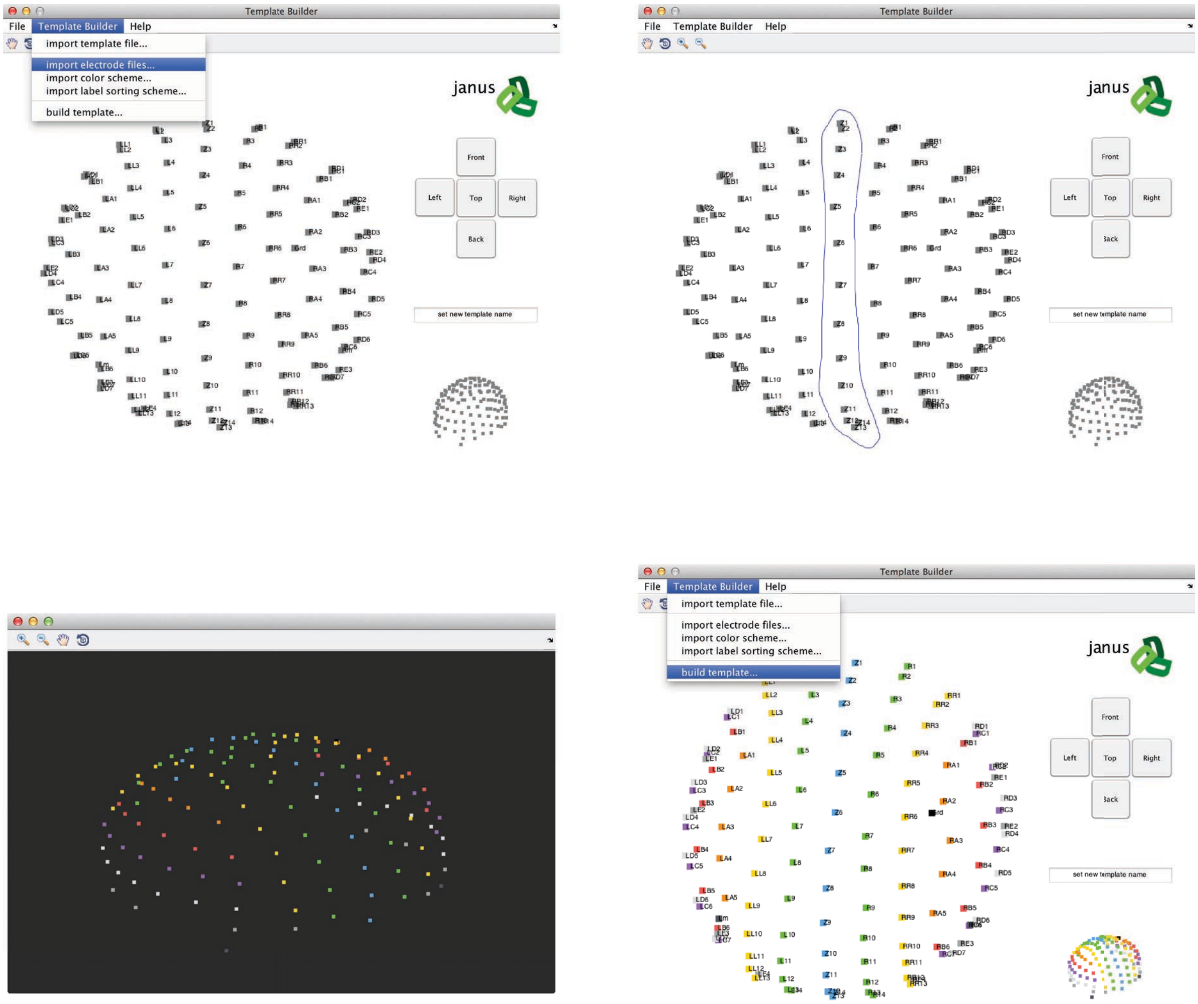

Figure 13: **Template Builder**

Example view on the janus3D Template Builder, depicting (top left) a loaded set of electrode positions, (top right) a drawn electrode selection for color specification, (bottom left) the 3D view window for the colored electrode set and (bottom right) an imported set of electrodes as it would be used to build the actual template file.

function to load a color scheme and apply it to a set of electrodes. Especially this is useful if electrode caps are provided in different sizes with the same respective labeling. Color schemes can be imported via the [Template Builder] → [import color scheme…] function and will be applied to the selected electrode files instantaneously. Furthermore the 'Template Builder' provides a function to sort template labels according to a specific order, which might be useful if using the guided electrode selection of janus3D. All electrode labels will be sorted according to the imported *.txt or *.mat file. Those files have to be formatted in n x 1 format using n as the respective label in the respective ascendant order. The mentioned function is accessible via [Template Builder] → [import label sorting scheme…]. A tabular description of the built template file is depicted in Table 4.

Table 4: Template Builder example structure

Structure Field	Annotation
type:	name chosen within the Template Builder
templates:	actual template structures
templates{filenumber}.points:	coordinates of electrode positions obtained from the model's surface
templates{filenumber}.label:	labels of coordinates obtained from the model's surface
templates{filenumber}.ColorCode:	respective color-code in [R,G,B] format. Values from 0 to 1.

10 Photo Masker

The 'Photo Masker' is a tool within janus3D to create masks for images either captured in front of a "greenscreen" or using a static background. It is accessible via [File] → [Photo Masker]. Selecting the folder where all images are stored, will directly load the first one. Images can only be imported in *.jpg format and will be stored as [original filename]_mask.png inside the mask folder. Clicking 'by color' enables the background selection based on the respective background color, which can be selected by framing a part of it. Double clicking the selection evokes the algorithm to define the threshold for the background removal. Using the slider bar at the sub-functions section allows adjusting the respective threshold. Per default 'Photo Masker' assumes the object to have a connection to one of the picture margins. Hence loading images, that are fully enclosed by the background, need to be specified accordingly by checking the 'enclosed' box. After clicking the 'create masks' button all images located in the image folder will be masked and stored in the respective mask folder, using the same respective threshold. It is also possible to create masks using a background image. Until so far only a single background image is supported. This background image should contain the exact same scenery but without showing the object. The algorithm subtracts the background from the image and creates the mask, based on that. 'Photo Masker' was designed to work hand in hand with Agisoft PhotoScan and hence creates mask files as they are required by PhotoScan. An example view on a 'Photo Masker' session, including an image and its corresponding mask can be found in Figure 14.

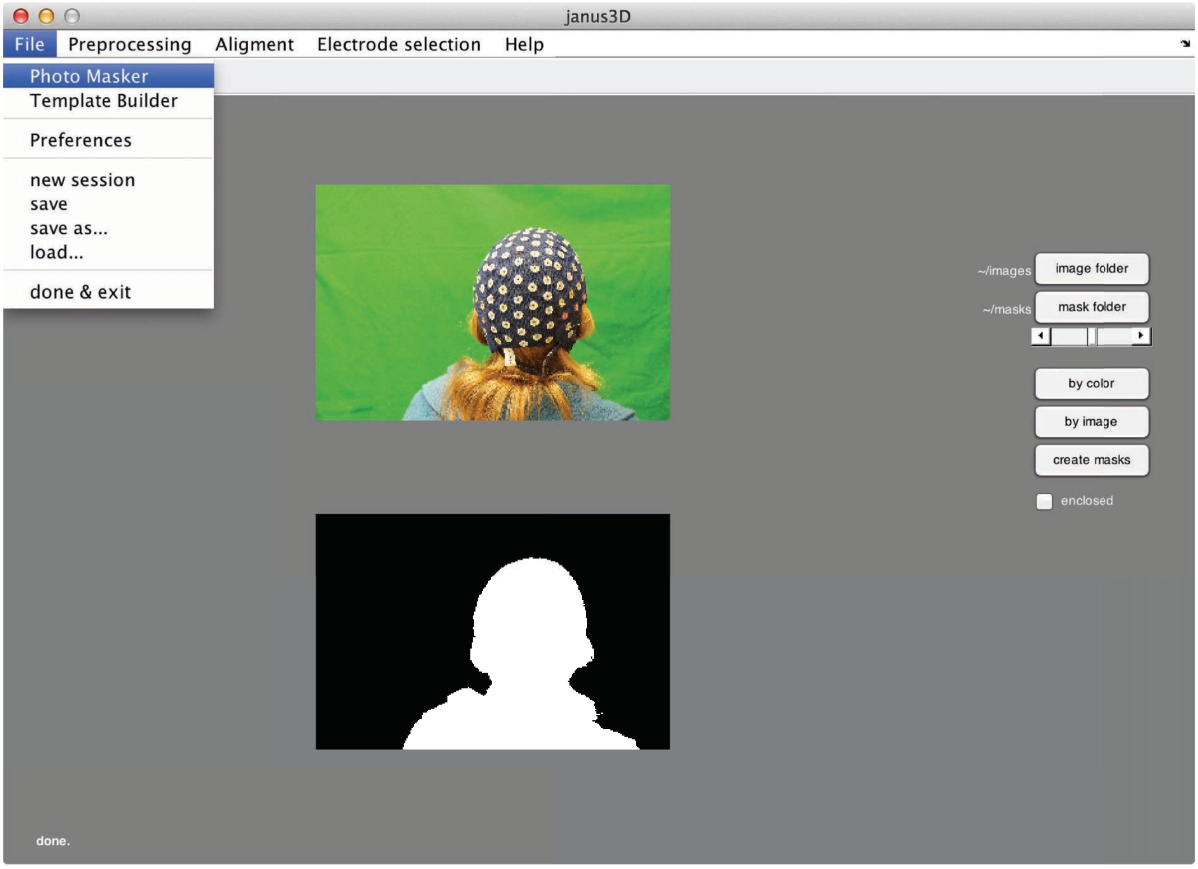

Figure 14: **Photo Masker**
Example view on the janus3D Photo Masker, depicting a loaded image and its corresponding mask that was created by color selection.

References

Besl, P. J., & McKay, N. D. (1992, April). Method for registration of 3-D shapes. In P. S. Schenker (Ed.), (pp. 586–606). doi: 10.1117/12.57955

Chen, Y., & Medioni, G. (1991). Object modeling by registration of multiple range images. In (pp. 2724–2729). IEEE Comput. Soc. Press. doi: 10.1109/ROBOT.1991.132043

MathWorks. (n.d.-a). *Computer Vision System Toolbox - MATLAB & Simulink - MathWorks Deutschland.* `http://de.mathworks.com/products/computer-vision/`.

MathWorks. (n.d.-b). *Image Processing Toolbox - MATLAB - MathWorks Deutschland.* `http://de.mathworks.com/products/image/`.

MathWorks. (n.d.-c). *MATLAB - MathWorks - MathWorks Deutschland.* `http://de.mathworks.com/products/matlab/`.

MathWorks. (n.d.-d). *Register two point clouds using ICP algorithm - MATLAB pcregrigid - MathWorks Deutschland.* `http://de.mathworks.com/help/vision/ref/pcregrigid.html`.

Oostenveld, R., Fries, P., Maris, E., & Schoffelen, J.-M. (2011). FieldTrip: Open Source Software for Advanced Analysis of MEG, EEG, and Invasive Electrophysiological Data. *Computational Intelligence and Neuroscience*, *2011*, 1–9. doi: 10.1155/2011/156869