Panoramic Imaging GUI – User's guide

The goal of the GUI PanoramicImaging_gui is to create a 3-D optical map with a realistic anatomy, specific to each studied heart.

Optical maps acquired by four cameras, placed approximately with an angle of 90° between them, are pre-processed with the GUI rhythm. The panoramic imaging GUI is used to perform three steps: a calibration of the panoramic imaging set-up (in other words, it calculates the position in the 3-D space of a point observed by a camera), a calculation of the silhouette of the heart and finally a projection of the optical maps acquired by the four different cameras onto the calculated silhouette. The GUI was designed to allow the user to perform these steps independently: the user can start from scratch by doing the three steps one after the other but if the calibration has already been done and the heart silhouette already been calculated, the user can choose to use these existing blocks and simply calculate the projection of a new optical map onto the silhouette. The user can also load already calculated 3-D maps to visualize them.

The GUI offers basic tools to visualize the resulting figure in MATLAB and to create simple movies to understand the 3-D activation pattern. However, the quality is limited and the options to manipulate the 3-D figure are scarce, I would advise to use another software (such as Amira) to generate good-looking figures and movies.

This user's manual first describes how to use the GUI. A second part provides help to understand the code (how to customize the different functions, how to fix some errors that may occur, how to improve some steps that are still operator dependent). A last part briefly explains the operations performed by this code.

1 – How to use the GUI

Launch Matlab. In the top panel, click on Set path (in the Environment part of the Home toolbar). Click on Add with subfolders to add the PanoramicImagingGUI folder and his subfolder geom3d to the Matlab path. Your current folder in Matlab can then be anything (either the PanoramicImagingGUI folder, the data folder or your regular Desktop folder) the calculations should be done the same. Once done, type PanoramicImaging_gui (case-sensitive) in the Command window and press enter. The GUI should open in a new window.

a) "Starting from scratch"

You need to first perform the calibration of the four cameras, i.e. calculate the position in the 3-D space of a point seen by a camera (or how to go from the 3-D space of the laboratory to the 2-D space of the camera).

Click on **Calibration Cam A** to select the picture of the dice taken by the camera A. Note that you can choose either .mat files (if you have already converted the camera files with the rhythm GUI) or .rsh files (the raw camera files). Then choose **Camera A** in the pop-up menu. A figure appears on the right panel showing you a reference picture with red numbers indicating the position of the reference points (i.e. remarkable points of known coordinates). Then click on **Load image and select points**. The picture of the dice taken by Camera A appears on the left panel. A red crosshair is displayed to help you select the 7 points of calibration (corners of the dice). In some situations, the selection of those points can be complicated by shadows or lights: you can use the two vertical sliders on the side of the figure to adjust the contrast of the image. In some situations, some frames acquired by the camera are corrupted: you can use the horizontal slider at the bottom of the panel to choose the reference frame. If you need to adjust the contrast or change frame, first select 7 random points, adjust the contrast or select a different frame and then click again on the **Load image and select points** button. Repeat this process for the four cameras. Once done, you can click on **Calculate calibration matrices** to calculate the four calibration matrices that will be saved in the initial data folder with the name *InitialFileNameCalibration* (InitialFileName being the name of the camera file MC051218-568 for example).

Secondly, you need to calculate the silhouette of the heart. Click on **Cam A 1st acq** to select the first picture of the heart taken by camera A. This picture is shown on the right panel. Click on **Cam A Last acq** to select the picture of the heart taken by camera A after the last rotation step. The picture is shown on the right panel. Make sure the angular step is set correctly. The threshold value can be manually adjusted. This value is used to perform an automatic segmentation of the heart: everything below this value is considered to be background, everything above is considered to be heart. To manually choose a value, in the top toolbar, click on the *Data Cursor button* and compare the index values of the background and of different points of the heart by clicking on different points of the right panel image. Enter the appropriate value in the Threshold box. Then click on **Calculate the silhouette**. Several steps are involved; a status bar will show you the progression of the calculation. The GUI will detect automatically if the .rsh camera files have already been converted to .mat files (if not, the conversion will be performed). After the first step of calculation, the user is asked to manually select, in 4 pictures, the rod holding the heart (i.e. the axis of rotation). The selection tool is a free-form tool that allows you to create a polygon by adding as many points as you want (just click to add a new point). To close the polygon, double click on the initial point (the pointer should change to a circle). Once all the steps of calculation performed, the resulting silhouette is displayed in 3-D in the bottom left panel. You can rotate the silhouette with the

Rotate 3D button  of the top toolbar. If you are satisfied with the silhouette, click **Save silhouette**. The corresponding data are saved in a .mat file (so you can re-use this silhouette for later calculations) and as a raw file (so you can visualize it in Amira, the data going from 0 to 1).

Finally, you can project optical maps calculated with the rhythm GUI onto the silhouette. Optical maps can be activation maps, dominant frequency maps, AP duration maps or phase movies. Click on **Optical Map Cam A** to select the map from camera A you would like to project onto the silhouette (the maps from other cameras will be loaded automatically) and then click on **Project optical maps onto the silhouette**. A pre-processing of the optical maps is performed: based on the previous segmentation, all optical map data located outside of the heart is removed. The four optical maps (one taken by each camera) are shown to the user prior to further calculation. The user is asked in a new window if he wants to remove any additional artifacts from the optical maps. If so, the user needs to select the camera corresponding to the map needing cleaning and then select a ROI by outlining the area of the optical map to KEEP. Repeat this operation for as many maps you want and then press **Cancel**. A window will pop-up to ask you if you plan to visualize this map in Amira later. The reason is that the direct calculation gives only a skeleton of heart and the rendering is not nice. If you want to visualize the map in Amira, an additional processing step will fill the gap and create a better rendering. However, this step is more time-consuming. Similarly, a reference map showing how the images for the four cameras are actually mapped onto the silhouette is calculated by default. This step is time consuming (especially if you want to calculate PhaseMovie) and can be skipped by desactivating the option on the right panel (**Ref map on/off**). The four maps are then automatically projected onto the silhouette. The resulting 3-D map is displayed on the bottom right panel and in a new figure. You can choose to display the map with/without a white background and the three axes (expressed in mm) and with/without a grid to help the reader visualize the dimensions. The vertical sliders allow the user to display only certain slices of the map: it is especially useful to mask the noisy data at the top and bottom of the map. You can choose the max value of the colormap in the right panel (for activation maps, the colormap goes from 0 to the max indicated, for phase movie it goes from –the max indicated to the max indicated). The user is invited to choose a format to save the figure (raster formats png, jpg and tiff as well as vector formats pdf and eps are offered, you can also save in .fig format to open it later with Matlab). You can save the data in a .mat file (Matlab format) to be able to load the map later by pressing the **Save data** button. A raw file is also automatically saved (if they did not exist already), it can be opened with Amira to create movies and animations. For the phase map movies, a new subfolder is created in the data folder and a raw file is created for each frame. Those files can be opened as a Time Series in Amira. You can visualize the 3-D activation pattern using the Rotate 3D button of the top toolbar or click on the Rotation Movie – **Play Movie** button for an automatic rotation around the heart. You can choose the **elevation angle** from which you see the heart during the rotation in the right panel. This movie can be saved as an .avi file for later visualization. If the optical map is a phase movie, the user indicates what frames need to be processed (by default the first ten frames). After all 3-D frames have been generated, the user can visualize the 3-D phase movie by clicking on Phase Movie – **Play Movie** or save it by clicking **Save movie.** The rotation and the phase movies cannot be combined in Matlab. Such elaborated animations are easier to create in Amira. The frame rates of all the movies can be adjusted in the right panel (**Movie frame rate**).

b) Starting with an existing calibration matrix

If the calibration has already been performed previously, the user can skip the reference point select step and directly load the existing calibration matrices by clicking on **Select an existing calibration file.** The name of the selected file is displayed in the Instructions box. You can then go to the next step: the calculation of the silhouette of the heart.

Click on **Cam A 1st acq** to select the first picture of the heart taken by camera A. This picture is shown on the right panel. Click on **Cam A Last acq** to select the picture of the heart taken by camera A after the last rotation step. The picture is shown on the right panel. Make sure the angular step is set correctly. The threshold value can be manually adjusted. This value is used to perform an automatic segmentation of the heart: everything below this value is considered to be background, everything above is considered to be heart. To manually choose a value, in the top toolbar, click on the *Data Cursor button*  and compare the index values of the background and of different points of the heart by clicking on different points of the right panel image. Enter the appropriate value in the Threshold box. Then click on **Calculate the silhouette**. Several steps are involved; a status bar will show you the progression of the calculation. The GUI will detect automatically if the .rsh camera files have already been converted to .mat files (if not, the conversion will be performed). After the first step of calculation, the user is asked to manually select, in 4 pictures, the rod holding the heart (i.e. the axis of rotation). The selection tool is a free-form tool that allows you to create a polygon by adding as many points as you want (just click to add a new point). To close the polygon, double click on the initial point (the pointer should change to a circle). Once all the steps of calculation performed, the resulting silhouette is displayed in 3-D in the bottom left panel. You can rotate the silhouette with the *Rotate 3D button*  of the top toolbar. If you are satisfied with the silhouette, click **Save silhouette**. The corresponding data are saved in a .mat file (so you can re-use this silhouette for later calculations) and as a raw file (so you can visualize it in Amira, the data going from 0 to 1).

Finally, you can project optical maps calculated with the rhythm GUI onto the silhouette. Optical maps can be activation maps, dominant frequency maps, AP duration maps or phase movies. Click on **Optical Map Cam A** to select the map from camera A you would like to project onto the silhouette (the maps from other cameras will be loaded automatically) and then click on **Project optical maps onto the silhouette**. A pre-processing of the optical maps is performed: based on the previous segmentation, all optical map data located outside of the heart is removed. The four optical maps (one taken by each camera) are shown to the user prior to further calculation. The user is asked in a new window if he wants to remove any additional artifacts from the optical maps. If so, the user needs to select the camera corresponding to the map needing cleaning and then select a ROI by outlining the area of the optical map to KEEP. Repeat this operation for as many maps you want and then press **Cancel**. A window will pop-up to ask you if you plan to visualize this map in Amira later. The reason is that the direct calculation gives only a skeleton of heart and the rendering is not nice. If you want to visualize the map in Amira, an additional processing step will fill the gap and create a better rendering. However, this step is more time-consuming. Similarly, a reference map showing how the images for the four cameras are actually mapped onto the silhouette is calculated by default. This step is time consuming (especially if you want to calculate PhaseMovie) and can be skipped by desactivating the option on the right panel (**Ref map on/off**). The four maps are then automatically projected onto the silhouette. The resulting 3-D map is displayed on the bottom right panel and in a new figure. You can choose to display the map with/without a white background and the three axes (expressed in mm) and with/without a grid to help the reader visualize the dimensions. The vertical sliders allow the user to display only certain slices of the map: it is especially useful to mask the noisy data at the top and bottom of the map. You can choose the max value of the colormap in the right panel (for activation maps, the colormap goes from 0 to the max indicated, for phase movie it goes from –the max indicated to the max indicated). The user is invited to choose a format to save the figure (raster formats png, jpg and tiff as well as vector formats pdf and eps are offered, you can also save in .fig format to open it later with Matlab). You can save the data in a .mat file (Matlab format) to be able to load the map later by pressing the **Save data** button. A raw file is also automatically saved (if they did not exist already), it can be opened with Amira to create movies and animations. For the phase map movies, a new subfolder is created in the data folder and a raw file is created for each frame. Those files can be opened as a Time Series in Amira. You can visualize the 3-D activation pattern using the Rotate 3D button of the top toolbar or click on the Rotation Movie – **Play Movie** button for an automatic rotation around the heart. This movie can be saved as an .avi file for later visualization. If the optical map is a phase movie, the user indicates what frames need to be processed (by default the first ten frames). After all 3-D frames have been generated, the user can visualize the 3-D phase movie by clicking on Phase Movie – **Play Movie** or save it by clicking **Save movie.** The rotation and the phase movies cannot be combined in Matlab. Such elaborated animations are easier to create in Amira. The frame rates of all the movies can be adjusted in the right panel (**Movie frame rate**).

c) Starting with existing calibration matrix and axis of rotation

If the calibration has already been performed previously, the user can skip the reference point select step and directly load the existing calibration matrices by clicking on **Select an existing calibration file.** The name of the selected file is displayed in the Instructions box. You can then go to the next step: the calculation of the silhouette of the heart.

If you have previously calculated a silhouette, you already have selected an axis of rotation. For some reason (you may be investigating the effect of a different angular step or of a different threshold), you need to re-calculate a silhouette. You can re-use the existing axis of rotation by selecting the xxxRotationAxis.mat file after clicking on **or an axis of rotation** button. Click on **Cam A 1st acq** to select the first picture of the heart taken by camera A. This picture is shown on the right panel. Click on **Cam A Last acq** to select the picture of the heart taken by camera A after the last rotation step. The picture is shown on the right panel. Make sure the angular step is set correctly. The threshold value can be manually adjusted. This value is used to perform an automatic segmentation of the heart: everything below this value is considered to be background, everything above is considered to be heart. To manually choose a value, in the top toolbar, click on the *Data Cursor button*  and compare the index values of the background and of different points of the heart by clicking on different points of the right panel image. Enter the appropriate value in the Threshold box. Then click on **Calculate the silhouette**. Several steps are involved; a status bar will show you the progression of the calculation. The GUI will detect automatically if the .rsh camera files have already been converted to .mat files (if not, the conversion will be performed). Once all the steps of calculation performed, the resulting silhouette is displayed in 3-D in the bottom left panel. You can rotate the silhouette with the *Rotate 3D button*  of the top toolbar. If you are satisfied with the silhouette, click **Save silhouette**. The corresponding data are saved in a .mat file (so you can re-use this silhouette for later calculations) and as a raw file (so you can visualize it in Amira, the data going from 0 to 1).

Finally, you can project optical maps calculated with the rhythm GUI onto the silhouette. Optical maps can be activation maps, dominant frequency maps, AP duration maps or phase movies. Click on **Optical**

Map Cam A to select the map from camera A you would like to project onto the silhouette (the maps from other cameras will be loaded automatically) and then click on **Project optical maps onto the silhouette**. A pre-processing of the optical maps is performed: based on the previous segmentation, all optical map data located outside of the heart is removed. The four optical maps (one taken by each camera) are shown to the user prior to further calculation. The user is asked in a new window if he wants to remove any additional artifacts from the optical maps. If so, the user needs to select the camera corresponding to the map needing cleaning and then select a ROI by outlining the area of the optical map to KEEP. Repeat this operation for as many maps you want and then press **Cancel**. A window will pop-up to ask you if you plan to visualize this map in Amira later. The reason is that the direct calculation gives only a skeleton of heart and the rendering is not nice. If you want to visualize the map in Amira, an additional processing step will fill the gap and create a better rendering. However, this step is more time-consuming. Similarly, a reference map showing how the images for the four cameras are actually mapped onto the silhouette is calculated by default. This step is time consuming (especially if you want to calculate PhaseMovie) and can be skipped by desactivating the option on the right panel (**Ref map on/off**). The four maps are then automatically projected onto the silhouette. The resulting 3-D map is displayed on the bottom right panel and in a new figure. You can choose to display the map with/without a white background and the three axes (expressed in mm) and with/without a grid to help the reader visualize the dimensions. The vertical sliders allow the user to display only certain slices of the map: it is especially useful to mask the noisy data at the top and bottom of the map. You can choose the max value of the colormap in the right panel (for activation maps, the colormap goes from 0 to the max indicated, for phase movie it goes from –the max indicated to the max indicated). The user is invited to choose a format to save the figure (raster formats png, jpg and tiff as well as vector formats pdf and eps are offered, you can also save in .fig format to open it later with Matlab). You can save the data in a .mat file (Matlab format) to be able to load the map later by pressing the **Save data** button. A raw file is also automatically saved (if they did not exist already), it can be opened with Amira to create movies and animations. For the phase map movies, a new subfolder is created in the data folder and a raw file is created for each frame. Those files can be opened as a Time Series in Amira. You can visualize the 3-D activation pattern using the Rotate 3D button of the top toolbar or click on the Rotation Movie – **Play Movie** button for an automatic rotation around the heart. You can choose the **elevation angle** from which you see the heart during the rotation in the right panel. This movie can be saved as an .avi file for later visualization. If the optical map is a phase movie, the user indicates what frames need to be processed (by default the first ten frames). After all 3-D frames have been generated, the user can visualize the 3-D phase movie by clicking on Phase Movie – **Play Movie** or save it by clicking **Save movie.** The rotation and the phase movies cannot be combined in Matlab. Such elaborated animations are easier to create in Amira. The frame rates of all the movies can be adjusted in the right panel (**Movie frame rate**).

d) Starting with existing calibration matrix and silhouette

If you already projected an optical map onto a silhouette, the calibration and silhouette calculation have already been performed. You can re-use these results to project a different type of optical maps.

Click **Select an existing calibration file** to load the existing calibration matrices**.** The name of the selected file is displayed in the Instructions box.

In the **Silhouette of the heart** panel, click on **Select an existing silhouette** to load the xxxSilhouette.mat file corresponding to your experiment. A quick calculation is performed to display the silhouette; the name of the selected file is indicated in the Instructions box.

Finally, you can project optical maps calculated with the rhythm GUI onto the silhouette. Optical maps can be activation maps, dominant frequency maps, AP duration maps or phase movies. Click on **Optical Map Cam A** to select the map from camera A you would like to project onto the silhouette (the maps from other cameras will be loaded automatically) and then click on **Project optical maps onto the silhouette**. A pre-processing of the optical maps is performed: based on the previous segmentation, all optical map data located outside of the heart is removed. The four optical maps (one taken by each camera) are shown to the user prior to further calculation. The user is asked in a new window if he wants to remove any additional artifacts from the optical maps. If so, the user needs to select the camera corresponding to the map needing cleaning and then select a ROI by outlining the area of the optical map to KEEP. Repeat this operation for as many maps you want and then press **Cancel**. A window will pop-up to ask you if you plan to visualize this map in Amira later. The reason is that the direct calculation gives only a skeleton of heart and the rendering is not nice. If you want to visualize the map in Amira, an additional processing step will fill the gap and create a better rendering. However, this step is more time-consuming. Similarly, a reference map showing how the images for the four cameras are actually mapped onto the silhouette is calculated by default. This step is time consuming (especially if you want to calculate PhaseMovie) and can be skipped by desactivating the option on the right panel (**Ref map on/off**). The four maps are then automatically projected onto the silhouette. The resulting 3-D map is displayed on the bottom right panel and in a new figure. You can choose to display the map with/without a white background and the three axes (expressed in mm) and with/without a grid to help the reader visualize the dimensions. The vertical sliders allow the user to display only certain slices of the map: it is especially useful to mask the noisy data at the top and bottom of the map. You can choose the max value of the colormap in the right panel (for activation maps, the colormap goes from 0 to the max indicated, for phase movie it goes from –the max indicated to the max indicated).The user is invited to choose a format to save the figure (raster formats png, jpg and tiff as well as vector formats pdf and eps are offered, you can also save in .fig format to open it later with Matlab). You can save the data in a .mat file (Matlab format) to be able to load the map later by pressing the **Save data** button. A raw file is also automatically saved (if they did not exist already), it can be opened with Amira to create movies and animations. For the phase map movies, a new subfolder is created in the data folder and a raw file is created for each frame. Those files can be opened as a Time Series in Amira. You can visualize the 3-D activation pattern using the Rotate 3D button of the top toolbar or click on the Rotation Movie – **Play Movie** button for an automatic rotation around the heart. You can choose the **elevation angle** from which you see the heart during the rotation in the right panel. This movie can be saved as an .avi file for later visualization. If the optical map is a phase movie, the user indicates what frames need to be processed (by default the first ten frames). After all 3-D frames have been generated, the user can visualize the 3-D phase movie by clicking on Phase Movie – **Play Movie** or save it by clicking **Save movie.** The rotation and the phase movies cannot be combined in Matlab. Such elaborated animations are easier to create in Amira. The frame rates of all the movies can be adjusted in the right panel (**Movie frame rate**).

e) Load an existing optical map

If you already created a 3-D optical map and you want to visualize the results or create new figures, simply press the **Load existing map** button. You can then manipulate this map as you wish.

2 – How to customize the GUI/understand the code

The GUI was built using the *guide* tool of Matlab. *Guide* is a visual interface to easily create GUIs. If you want to add new functions to the GUI or find some characteristics of the GUI, type *guide* in the command window of Matlab, click on the *Open existing GUI* tab and choose PanoramicImaging_gui. A new figure looking like the GUI opens. You can move the different elements, change their default values, add buttons, pop-up menus, graphs etc.

Each action on a button, menu etc. of the GUI has an effect that is coded in different functions called callbacks. For example, the pushbutton25 (which is the **Calculate the silhouette** button) is associated in the code with the function pushbutton25_Callback (it is a list of instructions and calls to other functions that perform the silhouette calculation). Comments have been added to the code to help the user to understand what function is performed by each callback.

a- How to customize the resolution and size of the final 3-D map

The 3-D heart is created by carving out an initial box based on the silhouettes observed by the different cameras at the different angles. The choice of the total size and the voxel size of this initial box has a direct effect on the resolution of the final 3-D map. The initial box is a cube of 40x40x40 mm^3. The voxel size is 0.2x0.2x0.2 mm^3. You can change these parameters (xmin, xmax, step X etc.) in the PanoramicImaging_gui.m code, in the pushbutton25_Callback (lines 777 to 785 of PanoramicImaging_gui.m).

The other parameter influencing the final resolution is the angular step used to analyze the optical maps. A step of 1° is used as default. You can change this value in the function ProjectionOptMap.m by changing the number of angular positions, nTheta (now set at 180).

b- How to customize the saving of figures

If you need to generate a figure with special requirements for a journal or a conference, you can customize the Save figure function (pushbutton34_Callback of PanoramicImaging_gui.m). The function that generates the figure is the *print* function. All details about the function can be read here:

https://www.mathworks.com/help/matlab/ref/print.html

The default resolution is 150DPI. To print the figure with a higher resolution, add the instruction '-rXXX', where XXX is your desired resolution to the print command of the lines 1073, 1075, 1077, 1081, 1082 of PanoramicImaging_gui.m.

c- How to customize the creation of movies

The saved movies present a frame rate of 15 frames per second and a video quality of 90%. You can adjust these default settings in the code (function pushbutton31_Callback for the rotation movie, lines 1014 and 1015, function pushbutton30_Callback for the phase movie, lines 954 and 955 of PanoramicImaging_gui.m).

You can also modify the way the camera rotates around the 3-D map. The position of the camera is coded by two angles: an elevation and a position around the z-axis, called azimuth. The code to define camera position is view(azimuth,elevation). The default rotation of the camera consists in a 360°-rotation around the z-axis at a given elevation of 30 degrees with a step of 1 degree. You can modify the line 1024 (in the function pushbutton31_Callback of PanoramicImaging_gui.m) to create a different movement of camera.

d- How to investigate the impact of angular step on the rendering of the silhouette

The impact of the angular step on the rendered silhouette was not investigated, a step of 5° was chosen based on the literature. It could be interesting to see if the rendered silhouette is acceptable with a bigger angular step (which would mean less experimental and computational time to spend).

To investigate that point, I would suggest generating the silhouette following the process described above. Once the silhouette is saved, rename it by Silhouette_AngularStep5. Modify the functions CalculateSilhouettesHeart.m and CalculateFinalSilhouette.m to increase the angular step between views: replace the line 18 (or 21 depending on the function) for iAngle = 1:**1**:numViews by for iAngle = 1:**2**:numViews to create a silhouette with an angular step of 10° (double of the previous one) or by for iAngle = 1:**3**:numViews to create a silhouette with an angular step of 15° (triple) etc. An interpolation might be necessary to obtain a smooth surface. With 5°, no interpolation is done: the points are so close to one another that they seem to be connected.

e- How to investigate the inter- and/or intra- operator dependency

The major source of variability of the final 3-D map resides in the very first step of calibration. Selecting different points of calibration creates a slightly different silhouette. If you are only interested in visualizing the activation patterns or in the relative positions of different points, this variability is not important. But if you want to know the absolute position of a specific point, you must understand that a certain error is attached to this position. The calibration step can be improved using more complex modelling of the camera and using the works of Tsai et al. *IEEE Journal of Robotics and automation*, 1987.

3 – Common errors occurring in the code and their meaning

Error using svd // Input to SVD must not contain NaN or Inf means that the axis of rotation was not selected correctly (most likely there were gaps or holes in the selected region). Please click again on **Calculate the silhouette** to re-do a new axis selection.

4 – Description of the operations performed in the GUI

The calibration is performed based on the works of McKay et al. *Comput Biomed Res,* 1982. The silhouette calculation step is based on the occluding contour method introduced by Niem et al. *SPIE Image and video proceedings,* 1994.

a – Calibration

A 3-D point (coordinates [x,y,z]) and its projected point in the image taken by camera (coordinates [u,v]) are related by the transformation matrix T and a scaling factor, k.

$$[x\ y\ z\ 1][T] = k\ [u\ v\ 1] \qquad \text{Eq. 1}$$

The *4x3* transformation matrix describes the perspective view; it can represent arbitrary linear operations. The assumption that the camera is geometrically linear is reasonable if the sensor array is linear and the lens of high quality.

To calibrate the camera A (i.e. to calculate the transformation matrix T), we use a reference object, a dice with dimensions similar to those of a heart (5.08x5.08x5.08 mm^3). The dice is tilted, held by a corner to present a 3-D profile to the cameras. Corners of the dice are used as calibration points. Each point must satisfy Eq. 1. The series of 3 linear equations can be reduced to 2 equations by substituting the scaling factor *k.*

$$(t_{11} - t_{13}u_i)x_i + (t_{21} - t_{23}u_i)y_i + (t_{31} - t_{33}u_i)z_i + (t_{41} - t_{43}u_i) = 0$$

$$(t_{12} - t_{13}v_i)x_i + (t_{22} - t_{23}v_i)y_i + (t_{32} - t_{33}v_i)z_i + (t_{42} - t_{43}v_i) = 0$$

where t_{ij} is the element of *[T]* located in the ith row and the jth column, (u_i, v_i) are the coordinates of the point in the image taken by camera A and (x_i, y_i, z_i) are the coordinates of the point in the 3-D space of the laboratory. If *n* calibration points are measured, the *n* pairs of equations can be represented in matrix form

$$[A][Tc] = t_{43}[B]$$

where [*A*] is a 2*n*x11 matrix, [*Tc*] is an 11-element column matrix containing 11 of the 12 elements of the transformation matrix [*T*] (t43 is the 12th element, it is a scale term and can be set to 1 without loss of generality) and [*B*] is a 2*n* element column matrix. The system presents 11 unknown parameters (the 11 elements of [*T*]). If at least 6 points are measured in the 2-D image, the system is overdetermined and can be solved using a generalized inverse:

$$[Tc] = ([A]^t[A])^{-1}[A]^t[B].$$

This operation is repeated for each of the four cameras.

NOTE: This calculation is performed in Matlab by the function called CalibratePhantom4OptMap4Cams.m.

b – Construction of a silhouette of the heart

The 3-D silhouette of the heart is built using the occluding contour methods. The method can be summarized in three steps. i) The images taken by each camera, are segmented to create 2-D silhouettes of the heart: the heart tissue is attributed a value of 1, the background a value of 0. ii) For each camera, a bounding pyramid using the calibration matrix and the 2-D silhouette. This step is similar to carving the 3-

D silhouette out of a virtual box. iii) The intersection of the four different bounding pyramids form the final approximation of the 3-D silhouette.

With four cameras, the resulting 3-D silhouette present only four faces and is a poor approximation of the real heart. The heart is mounted on a rotating platform. 18 additional views, with a step of 5°, are then acquired for each camera. The intersection of those 76 ((18+1)*4) bounding pyramids represents a more refined and more accurate approximation of the silhouette.

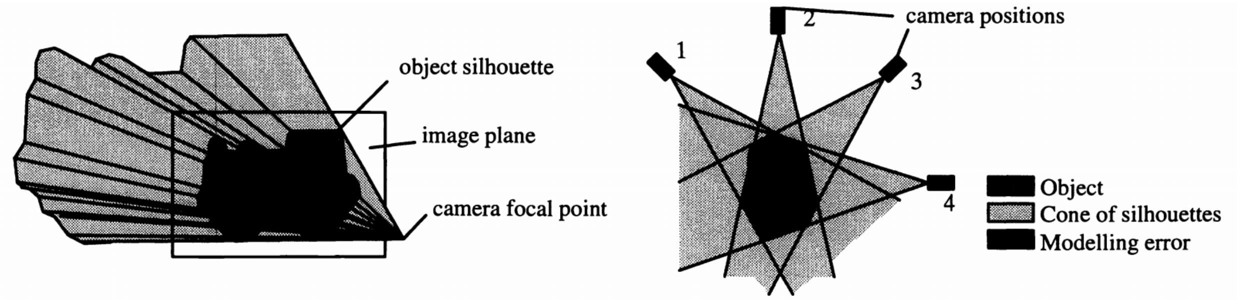

Fig. 4 : (a) Construction of a bounding pyramid, (b) Two dimensional description of the intersection of the bounding pyramids

From Niem et al. 1994

NOTE: The calculation of the 19 3-D partial silhouettes (with four faces) is performed in Matlab by the function called CalculateSilhouettesHeart.m. The axis of rotation of the heart is calculated based on the manual segmentation of the rod connecting the heart to the rotating platform (function CalculateAxisRotation.m). The 19 silhouettes are then rotated by the appropriate angle around the axis of rotation and added together to form the final silhouette (function CalculateFinalSilhouette.m). Because of the initial choice of the coordinate system (x,y,z are not the lateral, longitudinal and elevational directions of the laboratory but the directions of the dice, which is tilted), the resulting silhouette main axis is not the apex-base axis. The silhouette is rotated and translated to be at the center of the coordinate system and aligned with the z-axis. For display in Matlab, the positions of the wall of the silhouette are extracted in each slice (function DisplaySilhouetteHeart.m).

c – Projection of an optical map onto the silhouette

The 18 acquisitions done for the silhouette are not compatible with the high-frame rate required to observe the activation patterns of the heart. We decided to project the activation maps observed by the four cameras onto the silhouette obtained with a higher resolution.

First, the four optical maps are processed to create a 3-D optical map with four faces (each face corresponds to one of the optical maps). Finally, slice by slice, based on the angular position, the optical maps are projected onto the silhouette of the heart. The operation is simply a change of the radial position (from the optical map to the silhouette).

NOTE: The creation of the 3-D optical map is done in Matlab by the function PreprocessProjectionMap.m. Technically, a 4-faced silhouette is created using the occluding contour method, rotated and translated to be at the center of the system, aligned with the z-axis. In each z-slice, the rectangle is divided in 4 triangular areas so each face corresponds to the optical map observed with one of the cameras. The projection of this 3-D optical map onto the previously calculated realistic silhouette of the heart is performed by the function ProjectionOptMap.m

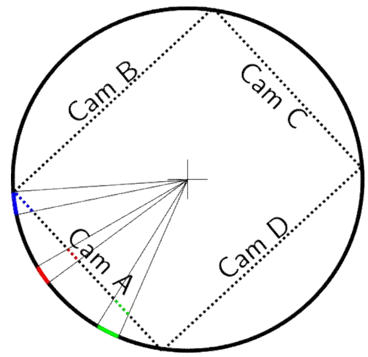

Figure 1Principle of the projection of the 4-faced optical map (dashed line) onto the more refined silhouette of the heart (bold solid line) for one slice. The value of the optical map read at an angular position ϑ is attributed to the point of the silhouette located at the same angular position.

5 – List of files

CalculateAxisRotation.m Calculate the axis of rotation of the heart using a manual segmentation of the rod performed by the operator on the initial images.

CalculateFinalSilhouette.m Combine the previously calculated *n* 3-D silhouettes (n=18 if you cover 90° with a 5° step) into a final smooth silhouette.

CalculateSilhouettesHeart.m Calculate the *n* 3-D silhouettes of the heart using the occluding contour method (n=18 if you cover 90° with a 5° step).

CalibratePhantom4Opt4Cams.m Calculates the 4 calibration matrices T, S, R and Q that describes the relation between the pictures taken by each camera and the 3-D experiment.

Calibration_LoadImages.m Convert if necessary and load the calibration images for a choosen camera.

CamACalReference.png, CamBCalReference.png, CamCCalReference.png, CamDCalReference.png Reference images depicting the position of the calibration points for each camera.

chooseCamDialog.m Function creating a dialog box asking the user if he wants to select only a ROI in the activation maps or use the whole map.

chooseSaveOptionDialog.m Function creating a dialog box asking the user if he wants to add a processing step to fill the gaps of the map for a better rendering in Amira.

CMOSConverter.m Convert the .rsh camera file into .mat file that can be used by Matlab.

DisplaySilhouetteHeart.m Rotate and translate the calculated silhouette in the center of the box aligned with the vertical axis. The surface of the heart is extracted to generate a plot with the function surf.

ginputRed.m Indicates the position of points selected by the user (as ginput.m) but the crosshair is red for easier selection.

JETColormap.am Colormap for visualization in Amira

PanoramicImaging_gui.fig Figure depicting the appearance of the GUI. Can be opened with the function guide in Matlab to easily perform modifications.

PanoramicImaging_gui.m Main code describing what function to run when a button of the GUI is pressed.

PreprocessProjectionOptMap.m Creates a 3-D map using each of the four optical maps taken by the cameras. This 3-D map does not look like a heart, it looks like a carved block of wood with 4 faces.

PreprocessProjectionOptMapPhaseMovie.m Same as above, but coded in a more efficient way to reduce the calculation costs when multiple frames are involved.

ProjectionOptMap.m Projects the 4-faced 3-D optical map (calculated with the PreprocessProjectionOptMap function) onto the realistic 3-D silhouette (calculated with the CalculateFinalSilhouette.m).

ProjectionOptMapPhaseMovie.m Same as above, but coded in a more efficient way to reduce the calculation costs when multiple frames are involved.

real2rgb.m Function used to read the camera files, converts a real value into a RGB value

rescale.m Function used by the real2rgb.m function

SmallDiceTilted_PierreCalibration.mat Matrices indicating the 3-D coordinates of the 7 reference points for each camera.

Geom3d Folder containing a series of functions used to perform geometrical calculations in 3-D (rotation, translation, fit etc.)

6 – For display in Amira

When the user press the Save figure in the Projection of the optical maps panel, a raw file with the projected 3-D map is saved. It can be visualized in Amira. Here are some tips for the visualization.

I created a colormap called JETColormap.am to reproduce the visual aspect of the Matlab figure. Because, the chosen resolution, the figure looks grainy in Amira (it is because each section is not a perfect smooth circle but a succession of little squares on a circular path). I found that you can attenuate that visual effect by adding a specular shading in Amira and by decreasing slightly the alpha scale to a value of 0.8 instead of 1. If you really want to get rid of that grainy/staircase effect, you need to change the resolution and go for a finer one (see section 2-a) how to change the resolution), but processing times will increase.

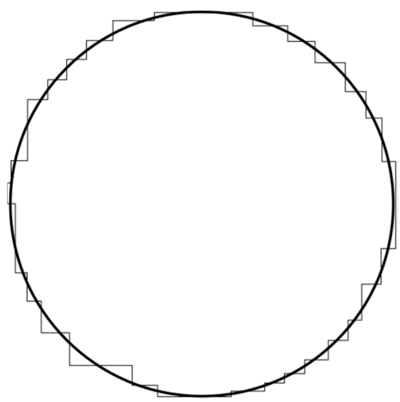

Illustration of the staircase effect, creating a grainy effect in Amira. The bold line indicates the ideal profile; the thin line indicates the calculated profile