Table of content

[Introduction 1](#_Toc40447625)

[Installation Prerequisite 2](#_Toc40447626)

[Python: 2](#_Toc40447627)

[CUDA: 2](#_Toc40447628)

[C++ 3](#_Toc40447629)

[Structure Projection: 3](#_Toc40447630)

[Compilation: 3](#_Toc40447631)

[User Guide: 3](#_Toc40447632)

[PyComView 5](#_Toc40447633)

[Compilation 5](#_Toc40447634)

[Service 5](#_Toc40447635)

[Compilation and testing 5](#_Toc40447636)

[Clinical 6](#_Toc40447637)

[Compilation and testing 6](#_Toc40447638)

[Installation on the iView computer 7](#_Toc40447639)

[User guide for ServiceApplication: 8](#_Toc40447640)

[Preparation of the calibration beams 8](#_Toc40447641)

[Gain Calibration: 8](#_Toc40447642)

[Panel Sag Calibration: 8](#_Toc40447643)

[User guide for Clinical Application: 9](#_Toc40447644)

[Viewing tools: 9](#_Toc40447645)

[Commissioning procedure: 10](#_Toc40447646)

[Troubleshooting py2exe compilation: 10](#_Toc40447647)

# Introduction

The fluoroscopy package includes four applications:

* **Structure Projection:** This application reads the DICOM files of the patient and generates the ROI projections that will be used by the **Clinical** application during the treatments.
* **Clinical:** This application is used during the treatments. It displays the beam’s eye view images of the patient with the ROI projections superimposed.
* **PyComView:** This application is used in background by the **Clinical** application and the **Service** application to communicate with the EPID and listen to the iCOM messages sent by the linac.
* **Service:** This application is used to generate the pixel gain calibration of the EPID and the panel position correction tables. The calibration files created are used by the **Clinical** application.

# Installation Prerequisite

The **Structure Projection** application can be run on any computer that has minimally a quad core CPU if the CPU version is used. The GPU version has only been tested on a Linux computer equipped with a NVIDIA GeForce GTX 1080. The **Clinical**, **Service** and **PyComView** applications are installed and used on the iView computer. However, you can compile them on another Windows computer before installing them on the iView.

Most of the sources files are python scripts. However, the GPU version of **Structure Projection** application uses one file that must be compiled with the NVIDIA CUDA compiler NVCC. Also, a part of the **PyComView** application must be compiled in c++. Follow these instructions for an easy installation.

## Python:

On a Windows computer connected to internet, first go to <https://www.python.org/> and install python-2.7.15, **32 bit** (use the Windows x86 MSI installer). Then, in a command prompt (cmd.exe), go to the directory: *C:\Python27\Scripts* and perform the following commands to install the necessary python packages:

1. pip install numpy==1.16.5
2. pip install matplotlib==2.2.5
3. pip install wxPython==4.0.7.post2
4. pip install pyautogui==0.9.48
5. pip install scipy==1.2.2
6. pip install PyOpenGL==3.1.0
7. pip install dicom==0.9.9.post1
8. pip install skimage==0.14.1
9. pip install Pillow==4.0.0
10. Go to <https://www.py2exe.org/> Download and execute the py2exe installer for python2.7

Please note that even if only these specific library versions have been tested, we expect the programs to run correctly with other versions of these libraries.

## CUDA:

The GPU version of the **Structure Projection** application was tested on a Linux computer only. We used the CUDA Toolkit version 9.2 and the driver version 396.54. We recommend that you follow the NVIDIA CUDA installation procedures: <https://developer.nvidia.com/cuda-toolkit-archive> .

## C++

Go to: <http://www.mingw.org/> and download the MinGW Installation Manager (mingw-get-setup.exe). Launch the application and select *mingw32-base-bin*, *mingw32-gcc-g++-bin* (version 8.2.0) and *msys-base-bin* packages to install. Once selected, click Installation🡪Apply Changes. Figure 1 present a screenshot of the MinGW Installation Manager. Assuming MinGW was installed in C:\MinGW\, the following paths need to be added to your Windows PATH environment variable: C:\MinGW\bin\ and C:\MinGW\msys\1.0\bin .

|  |
| --- |
|  |
| Figure 1. Screenshot of the MinGW Installation Manager with the packages selected |

# Structure Projection:

## Compilation:

The CPU version of **Structure Projection** is a python script that doesn’t need to be compiled before executing. For the GPU version, one compilation must be done. Be sure that you have already installed the CUDA toolkit. In a command prompt, go to the repository /StructureProjection/StructureProjection/. Then type the command: “make”. The following file should be created: StructureProjection/StructureProjection/main .

## User Guide:

Before using the **Structure Projection** application, you have to make sure that the file settings.txt in the folder /StructureProjection/ contains the good parameters. The first parameter is **DefaultPATHScanFolder** and it is simply equal to the full path of the folder that will be open by default in the browser that pops when you click on the button ***Scan Folder***. The second parameter is **SavingPath** and it must be equal to the full path of the folder where the results of the projections will be saved. This folder is common to all patients and it must be accessible by the iView computer. The last parameter **GPU** determines if you want to compute on the GPU or not and it must be equal to either True or False.

Follow this step-by-step procedure to project structures:

1. In the TPS export the RTPLAN, the RTSTRUCT and the CT images in an empty folder. Because the software is not entirely robust and to avoid potential errors, only one dicom serie of images should be exported and the structures you want to project must have been drawn on this serie.
2. Launch GUI.py (see figure 2)
3. Click on the *Scan Folder* button. In the browser, select the folder in which you exported the DICOM files.
4. Select the prescription you want in the scroll-down menu.
5. Click on the checkboxes of the ROIs you want to project.
6. In the text box, write the name (without space) you want to give to the file containing the projections.
7. Click on the *Project Structures* button and wait.
8. Once the computation is completed, you can visualize the projections in function of the gantry angle with the scrollbar.
9. Click the X at the top-right corner of the window to close the application.

If only the CPU is used, it can take a very long time before the computation is completed. The option to compute with the CPU is only there so that you can try the software even if you don’t have a NVIDIA GPU card. But it is highly suggested to use the GPU computation if you want to routinely use this software in your clinic.

|  |  |
| --- | --- |
| A) | B) |
| Figure 2: GUI of **Structure Projection** A) Select the ROIs you want to project B) After the projection computation is completed, verify the ROI projections computed in the EPID reference frame. | |

# PyComView

In PyComView, there is a file config.txt which contains the variable ip\_iCOM, which is the ip address of the Integrity TCS computer (linac) in the NSS. By default, this variable is set to: 192.168.30.2. You can change this value.

## Compilation

Before compiling the c++ part of **PyComView**, You first need to obtain some files:

1. From <https://github.com/esrf-bliss/Lima-camera-perkinelmer/tree/master/sdk/msvc/include>, get the file Acq.h and insert it in the folder : \PyComView\iCom\_iView\inc\
2. From <https://github.com/esrf-bliss/Lima-camera-perkinelmer/tree/master/sdk/msvc/bin>, get the file XISL.lib and insert it in the folder \PyComView\iCom\_iView\src\
3. From the software iCOMCAT 13.00 (Elekta Limited) that should already be installed on your iView computer,
   1. get a copy of the file iCOMAPI.h and insert it in folder: \PyComView\iCom\_iView\inc\
   2. get a copy of the file iCOMPClient.lib and insert it folder : \PyComView\iCom\_iView\src\

Start a command prompt (cmd.exe) and go to the folder \PyComView\iCom\_iView\. Then, use "make" as command line to compile the C++ code. This should create the file \PyComView\iCom\_iView\src\iCom\_iView.dll

For compiling the python part of **PyComView**, execute the file \PyComView\compile.bat by double clicking on it, assuming python.exe is in C:\python27\python.exe, otherwise the compile.bat can be modified accordingly. The folder \dist\ should be created in the folder \PyComView\. In addition, a few library files should be added to the \PyComView\dist\ folder:

1. From the iCOMCAT 13.00 (Elekta Limited), get a copy of the file iCOMClient.dll and insert it in folder : \PyComView\dist\
2. From <https://github.com/esrf-bliss/Lima-camera-perkinelmer/tree/master/sdk/msvc/bin>, get a copy of the file XISL.dll and insert it in the folder \PyComView\dist\
3. From \MinGW\bin\ get a copy of the files libgcc\_s\_dw2-1.dll and libstdc++-6.dll and insert a copy in folder : \PyComView\dist\
4. If not copied by the compile.bat, get the file \PyComView\iCom\_iView\src\iCom\_iView.dll and insert a copy in the folder : \PyComView\dist\
5. If not copied by the compile.bat, get the file \PyComView\config.txt and insert a copy in the folder: \PyComView\dist\

# Service

## Compilation and testing

In a command prompt, go inside the folder \Service\_SourceFiles\ and perform the command: *python setup.py py2exe*. If the operation is successful, two folders should be created: build and ServiceApplication. The executable is the file ServiceApplication.exe in the folder \Service\_SourceFiles\ ServiceApplication\.

You can test the executable ServiceApplication.exe on the Windows computer you used for compilation (not necessarily the iView computer). First, create a folder name EPID on your *C* hard drive: C:\EPID\. Then, get a copy of the file \FluoMV\settings.txt and place it inside the folder: C:\EPID\. Makes sure the variable: DebugLevel=2079 in settings.txt. Then, double click on ServiceApplication.exe, you should see the following window displayed in figure 3. If you encounter problems with the executables created with py2exe, please check the section Troubleshooting py2exe compilation.

|  |  |  |
| --- | --- | --- |
| |  | | --- | |  | | Figure 3. Graphical user interface of ServiceApplication.exe | |
|  |

# Clinical

## Compilation and testing

In a command prompt, go inside the folder \Clinical\_SourceFiles\ and perform the command: *python setup.py py2exe*. If the operation is successful, two folders should be created: build and ClinicalApplication. The executable is the file ClinicalApplication.exe in the folder \Clinical\_SourceFiles\ ClinicalApplication\.

You can test the executable ClinicalApplication.exe on the Windows computer you used for compilation (not necessarily the iView computer). If not already done, first create a folder name EPID on your *C* hard drive: C:\EPID\. Then, get a copy of the file \FluoMV\settings.txt and place it inside the folder: C:\EPID\. Makes sure the variable: DebugLevel=2079 in settings.txt. Then, double click on ClinicalApplication.exe, you should see the following window:

|  |  |  |
| --- | --- | --- |
| |  | | --- | |  | | Figure 4: Graphical user interface of ClinicalApplication.exe when testing the compilation | |
|  |

# Installation on the iView computer

Follow these steps for installing the Fluoroscopy software on the iView computer. Please note that the iViewGT software must always be closed when using the FluoroscopyMV software.

1. Create a folder named EPID on the *C* hard drive of the iView computer: C:\EPID.
2. Get a copy of the file \FluoMV\settings.txt and place it inside the folder: C:\EPID.
3. Makes sure the variables of settings.txt are correctly configured:
   1. roi\_path: Path of the folder containing the results of the ROI projections of the patients
   2. SavePNG: Set this variable to True if you want the images of the **Clinical** application to be saved in png. When you close the **Clinical** application, the images will be saved in the folder: \roi\_path\ SavedImages\.
   3. SaveRaw: Set this variable to True if you want the raw images of the Clinical application to be saved in float 32 bits (size=512x512 pixels). When you close the **Clinical** application, the images will be saved in the folder:

\roi\_path\ SavedImages\.

* 1. DebugLevel=0, 1 or 2. We recommend using 1.
  2. EqualizeHistogram=True or False. In clinical situation, this parameter should always be equal to True.

1. Copy the folder \PyComView\ with the folder \dist\ inside and place it in C:\EPID\.
2. Copy the folder ServiceApplication and place it inside C:\EPID\.
3. Copy the folder ClinicalApplication and place it inside C:\EPID\.
4. At the end, the folder C:\EPID\ should have the structure presented at figure 5.

|  |
| --- |
|  |
| Figure 5. Screenshot of the C:\EPID\ folder on the iView computer before executing ServiceApplication.exe |

# User guide for ServiceApplication:

**ServiceApplication.exe** should be the first program to be executed on the iView computer. It is used to perform the gain calibration of the pixels of the EPID and to create the panel position correction tables. The panel must always be at its central position.

## Preparation of the calibration beams

We recommend creating and delivering the calibration beams with the iCOM Customer Acceptance Test (iCOMCat.exe) software provided by Elekta. This software should already be installed on the iView computer. The control points of the beams are specified in the file: \FluoMV\iCOMCAT\_CalibrationSequences.xlsx.

## Gain Calibration:

1. Be sure that iViewGT.exe is closed.
2. Launch iCOMCat.exe and load the calibration beam GainCalibration6MV.efs that you have previously created
3. Launch ServiceApplication.exe
4. Click on the button BACKGROUND in the GAIN CALIBRATION section. Once the background acquisition is completed, the START button will enable.
5. Click on the button START
6. Start the irradiation of the calibration beam GainCalibration6MV.efs
7. Once the beam is finished, click on the STOP button. After a few seconds, a message confirming that a new Gain Calibration has been saved should appears.

## Panel Sag Calibration:

1. In the treatment room, install the ballbearing provided by Elekta to the treatment table and bring it to the radiation isocenter with the kvFlexmap beam (same procedure as for the flexmaps of the XVI).
2. This procedure is in two steps because the panel sag calibration requires a clockwise and a counterclockwise gantry rotation.
3. With iCOMCat.exe, load the clockwise panel sag calibration beam PanelSagCalibration6MV\_CW.efs that you have previously created.
4. Launch ServiceApplication.exe
5. Click on the button BACKGROUND in the PANEL SAG CALIBRATION section. Once the background acquisition is completed, the START button will enable.
6. Click on the button START
7. Start the irradiation of the beam PanelSagCalibration6MV\_CW.efs
8. After a few seconds, you should see the panel sag calibration figure to update in real time
9. Once the beam is finished, click on the STOP button. After a few seconds, a message confirming that a new CW Panel Position Correction Table has been saved should appears.
10. Close the application and reopen it
11. With iCOMCat.exe, load the counterclockwise panel sag calibration beam PanelSagCalibration6MV\_CC.efs that you have previously created.
12. Click on the button BACKGROUND in the PANEL SAG CALIBRATION section. Once the background acquisition is completed, the START button will enable.
13. Click on the button START
14. Start the irradiation of the beam PanelSagCalibration6MV\_CC.efs
15. After a few seconds, you should see the panel sag calibration figure to update in real time
16. Once the beam is finished, click on the STOP button. After a few seconds, a message confirming that a new CC Panel Position Correction Table has been saved should appears.

# User guide for Clinical Application:

1. Be sure that iViewGT.exe and ServiceApplication.exe are closed.
2. Launch ClinicalApplication.exe
3. Enter the PatientID and press Search. This assumes that the projections are already computed and saved in roi\_path (in settings.txt).
4. Select the prescription corresponding to the treatment
5. Click on Background
6. When the beam start, press Start
7. Once the treatment is completed, press Stop
8. Close the application after each treatment by clicking on the X. This will start the copy of the images if you set the SavePNG or SaveRAW to True in settings.txt.

Please note that if the acquisition freezes during treatment, press Stop and then Start to restart the acquisition.

## Viewing tools:

* Right mouse button click on the viewing window switches between the Zoom and Drag tools
  + In drag mode, hold the left mouse button pressed and move the cursor to drag the image
  + In zoom mode, hold the left mouse button pressed and move up to zoom in and down to zoom out.
* The sliding bar below the image sets the threshold used by the automatic histogram equalization. All pixels with a value below this threshold on the raw image are put to 0 and the histogram equalization is performed on all the pixels with a value above the threshold.
* When you click Stop, you can review the images previously acquired by using the mouse wheel.

# Commissioning procedure:

We recommend that you read the article “An open source software for monitoring intrafraction motion during external beam radiation therapy based on superimposition of contours of projected ROIs on cine-MV images” in the Journal of Applied Clinical Medical Physics before proceeding to the commissioning of the FluoroscopyMV software.

1. Acquire a CT scan of a ball bearing phantom with a slice thickness of 1.0 mm.
2. Import the CT images in a treatment planning system
3. Create a ROI by contouring the ball
4. Create a simple treatment plan with an arbitrary number of monitor units that contains a beam with the isocenter positioned with an offset of your choice lateraly, longitudinally and vertically. MLC should be turned on.
5. Export in dicom the CT images, the RTSTRUCT and the RTPLAN files in an empty folder.
6. Launch **Structure Projection** to generate the ROI projections.
7. In the treatment room, place the BB according to the treatment plan isocenter.
8. On the iView computer, set the variable EqualizeHistogram=False in the file C:\EPID\settings.txt
9. Launch **ClinicalApplication.exe** and select the prescription of your treatment plan.
10. Perform acquisitions with an arc beam of field size 24×24cm2 and verify the coincidence between the BB and the ROI projections. Different gantry speeds should be tested for the two directions of rotation.

# Troubleshooting py2exe compilation:

Compiling python scripts into Windows executables using py2exe can sometimes be frustrating. Depending on the version of the Windows operating system used for compiling the applications, errors related to module importation and *dll* loading may occur when the executable is launched from the iView computer. This section contains tips to help compile the executables.

First, make sure you have installed the specific version of python and python libraries mentioned in the section **Installation Prerequisite**. Second, before compiling an application with py2exe again, be sure to delete the folders created by py2exe: build and ServiceApplication (or ClinicalApplication or dist), before running the command *python setup.py py2exe* again.

Figure 6 shows the setup.py file of ServiceApplication. During our tests, we encountered two types of errors, both related to *dll* files. First, an error message can occur when executing the command: *python setup.py py2exe* (see Figure 7). This type of error can be resolved by adding the name of the *dll* file to the dll\_excludes list in the setup.py file. In our case, we had to add the file MSVCP90.dll to the dll\_excludes list in setup.py (see Figure 6).

The second type of error occurred when we launched the executable ServiceApplication.exe from the iView computer. The software did not start and an error message appeared. Then, a file named **ServiceApplication.exe.log** was created in the folder C:\EPID\ServiceApplication which contained the error message shown at Figure 8. After trial and error, we found that it was caused by the file *OLEACC.dll* placed in the folder ServiceApplication by py2exe. This type of error can be resolved by deleting the problematic *dll* file from the ServiceApplication folder or preferably by adding the name of this file to the dll\_excludes list in setup.py. Depending on the version of the Windows operating system used when compiling the python scripts, different *dll* files can be problematic. If you encounter problems, we therefore recommend that you check the list of *dll* files in the ServiceApplication folder created when executing the command: *python setup.py py2exe*. Figure 9 shows two examples of list of *dll* files contained in the ServiceApplication folder for two different compiled versions that worked correctly on the iView computer. We expect that the list of *dll* files in your compiled folder ServiceApplication should not be much different from the lists shown in Figure 9. The same logic applies for ClinicalApplication. The figures 10 and 11 present the lists of *dll* files generated when compiling ClinicalApplication on two different computers. In both cases, the executable worked correctly on the iView computer.

|  |
| --- |
|  |
| Figure 6. The setup.py file of ServiceApplication. Editing the dll\_excludes list may be necessary. |

|  |
| --- |
|  |
| Figure 7. Error message encountered when executing the command:  *python setup.py py2exe* |

|  |
| --- |
|  |
| Figure 8. Error message of the file ServiceApplication.exe.log created when trying to launch ServiceApplication.exe on the iView computer. |

|  |  |
| --- | --- |
| **Example 1** | **Example 2** |
|  |  |
| Figure 9. Two examples for the list of *dll* files contained in the ServiceApplication folder when the application is compiled on two different computers. In both cases, the executable worked correctly on the iView computer. | |

|  |
| --- |
|  |
| Figure 10. An example of the list of all the *dll* files contained the ClinicalApplication folder. |

|  |
| --- |
|  |
| Figure 11. Same as Figure 10 but for a compilation on a different computer. |