

SDK Rapid Scan Demo Description Document

Check whether the device is powered and properly connected before scanning. Note that the USB interface should be connected to the USB3.0 interface of the computer. The following is the operating instructions of the SDK Rapid Scan Demo. The process is shown in figure 1.

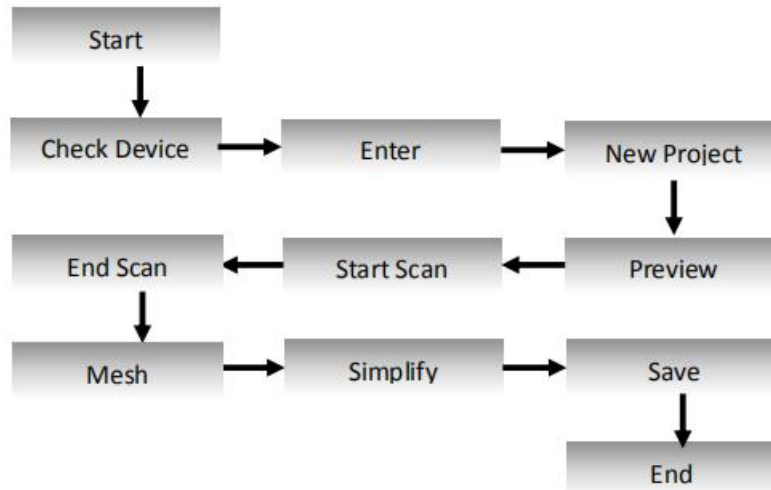


Figure 1 Process of SDK Rapid Scan Demo

1、Start

Open the SDK Rapid Scan, the interface is shown in figure 2.

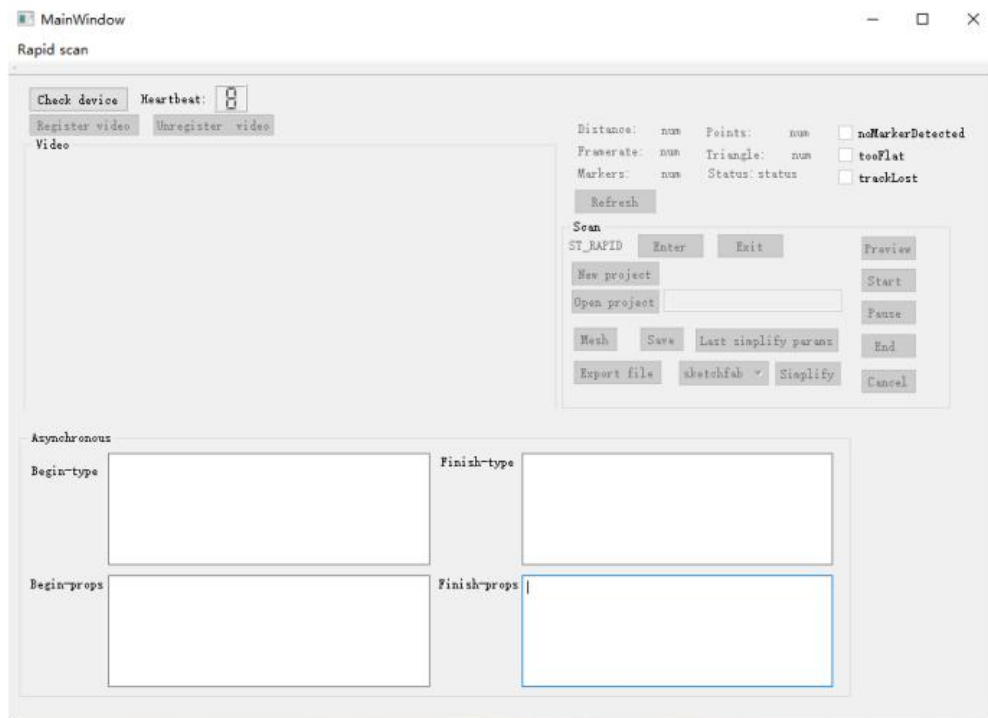


Figure 2 SDK Rapid Scan Interface

2、Check Device

As shown in figure 3, click the "check device" button to enter the detection status of the device, and a progress bar indicates the progress of the detection.

Corresponding functions: mianwindow.cpp->on_pushButton_DeviceCheck_clicked

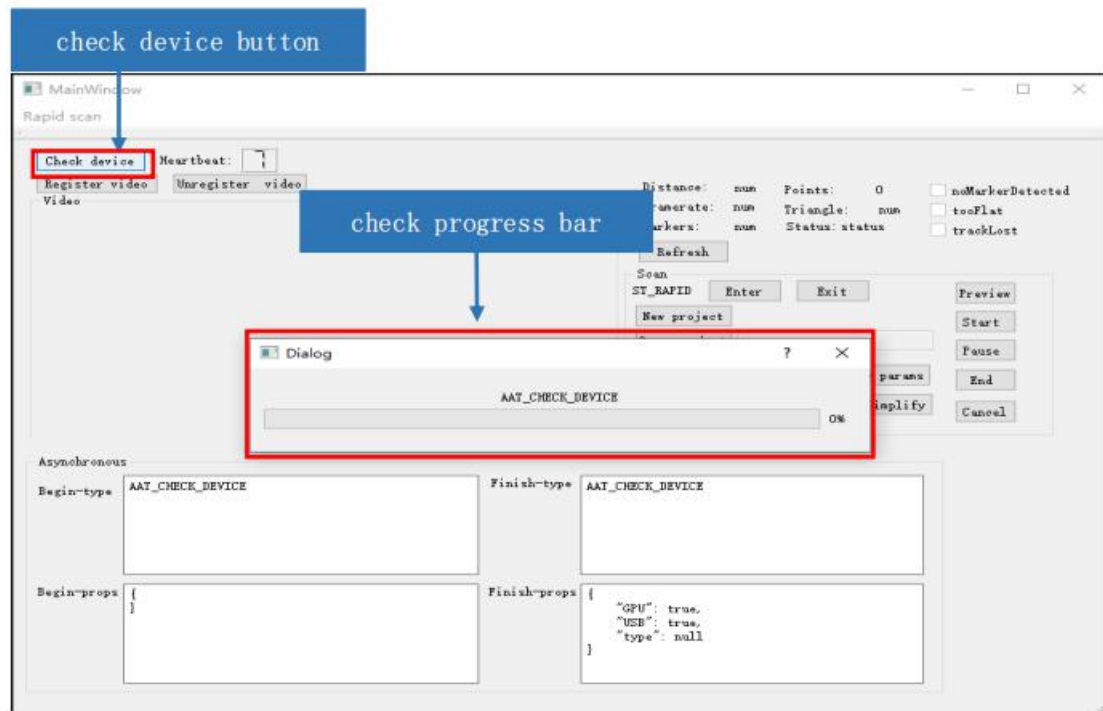


Figure 3 "Check device" button interface diagram

3、Enter

Click the "Enter" button to enter the scanning state, and the interface will display a progress bar, as shown in figure 4.

Corresponding functions: mianwindow.cpp->on_pushButton_ScanEnterScan_clicked

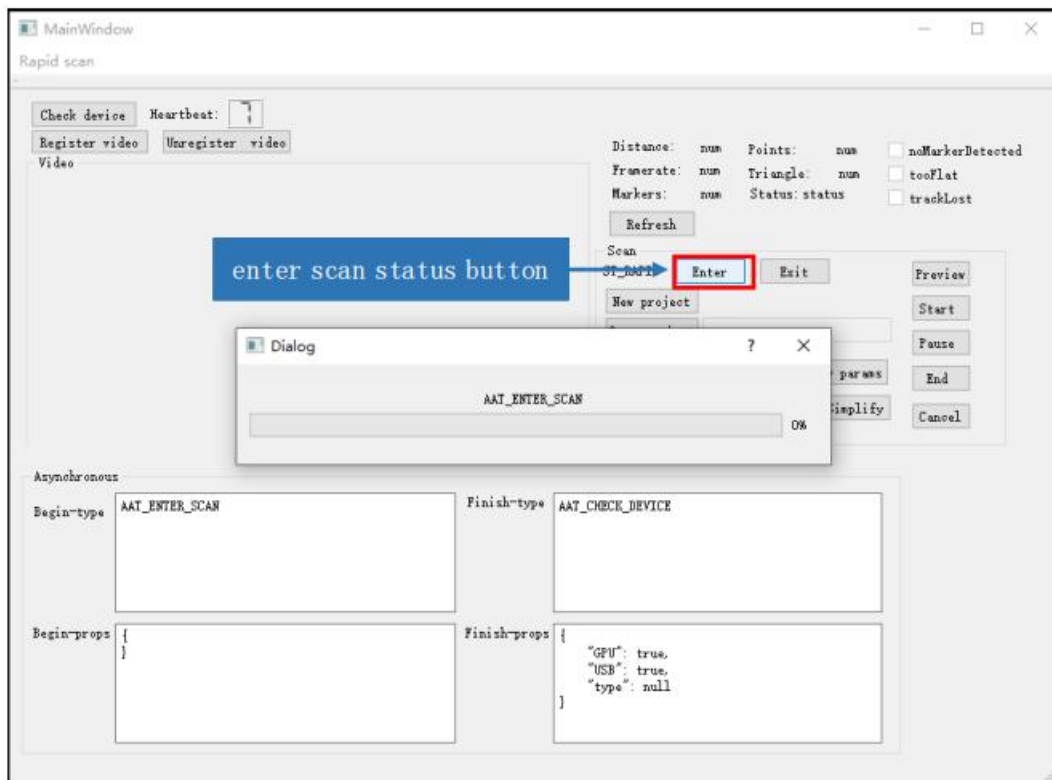


Figure 4 "Enter" button interface diagram

4、New Project

Click the “New Project” button to enter the new project interface as shown in figure 5. The parameters of the new project are shown in table 1.

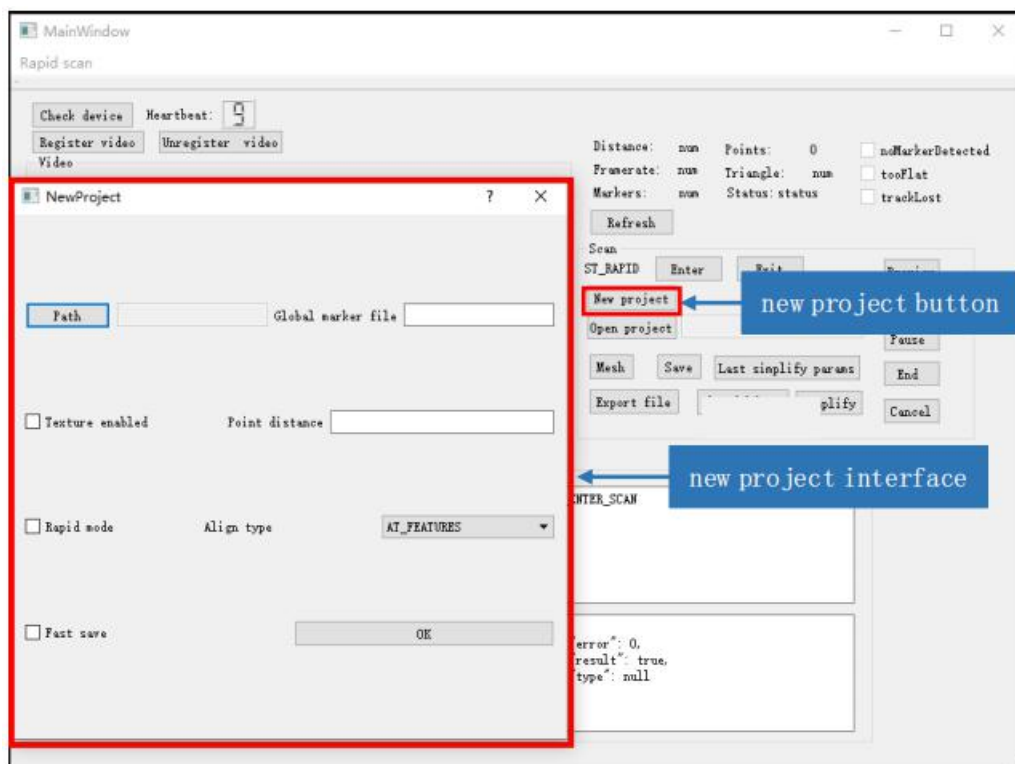


Figure 5 New Project interface diagram

Table 1 New project parameters

Parameter	Implication	UI Type	Function
Path	Select the path to save the project	Button+Text	commonui.cpp->onPathButtonClicked
Global marker file	Global Framepoint File Path	Button+Text	commonui.cpp->NewProject(QDialog *parent)
Texture enabled	Whether to use a texture camera	Check box	commonui.cpp->NewProject(QDialog *parent)
Rapid mode	Determine the scan frame rate	Check box	commonui.cpp->NewProject(QDialog *parent)
Fast save	Whether to save each frame of image	Check box	commonui.cpp->NewProject(QDialog *parent)
Point distance	Point distance	Text	commonui.cpp->NewProject(QDialog *parent)
Align type	Scan type: AT_FEATURES(Feature stitching) AT_MARKERS(Mark stitching) AT_HYBRID(Mixed stitching) AT_AUTO(Automatic stitching)	Drop-down list	commonui.cpp->NewProject(QDialog *parent)
ok	Confirm entry to new project	Button	commonui.cpp->NewProject(QDialog *parent)

We choose the path of the new project, set the point distance to 1, the other is the default, click the "ok" button. The interface of the new project is shown in figure 6. The parameters of the main scanner in the upper right corner are shown in table 2.

When the rapid mode and Fast save check boxes are selected, the speed of the point cloud increases faster during scanning. The frame rate is twice as high as when the Rapid mode check box is unchecked, and the final exported file is relatively large.

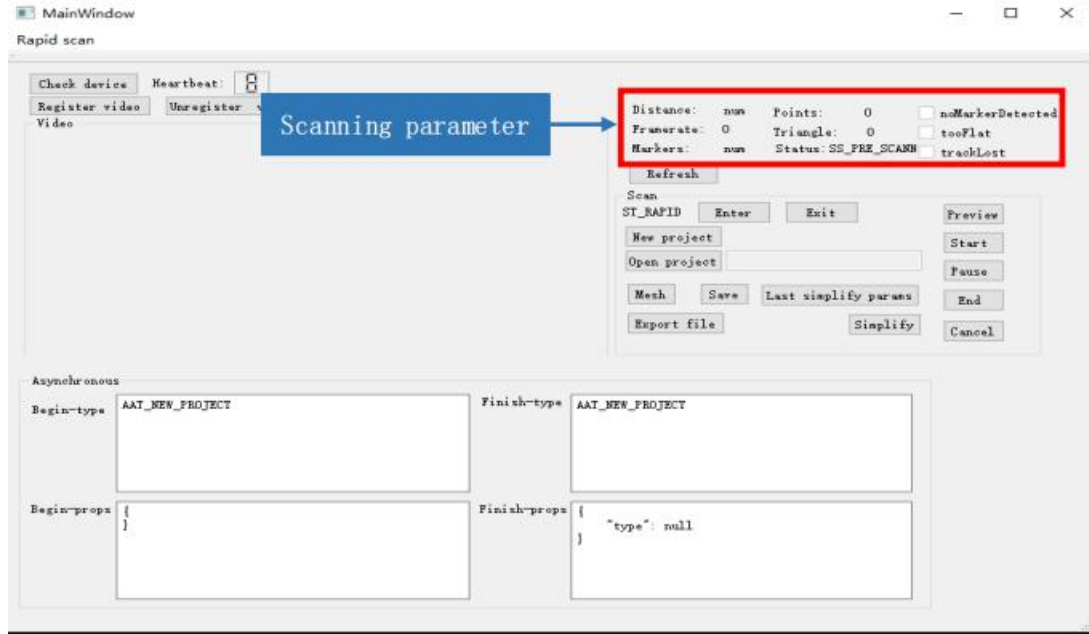


Figure 6 Scanning parameters

Table 2 Scanning parameters

Parameter	Implication	UI Type	Function
Distance	Distance between the device and the scanned object	Text	mainwindow.cpp-> ScanDist
Framerate	Scan frame rate	Text	mainwindow.cpp-> ScanFramerate
Markers	Number of point clouds scanned	Text	mainwindow.cpp-> ScanMarkerCount
Points	Model triangle face number	Text	mainwindow.cpp-> ScanPointCount
Triangle	Device scan status	Text	mainwindow.cpp-> ScanTriangleCount
Status	No scan to mark point (Cannot check manually, automatically judge)	Check box	mainwindow.cpp-> ScanStatus
noMarkerDetected	Too smooth (Cannot check manually, automatically judge)	Check box	mainwindow.cpp-> ScanNoMarkerDetected
TooFlat	Tracking loss (Cannot check manually, automatically judge)	Check box	mainwindow.cpp-> ScanTooFlat
trackLost	Tracking loss (Cannot check manually, automatically judge)	Check box	mainwindow.cpp-> ScanTrackLost

5、Preview

Click the “Preview” button to enter the device pre-scan mode. The interface is shown in figure 7. The parameters of the pre-scan interface are shown in table 3.

Corresponding functions: mianwindow.cpp->on_pushButton_Pre_clicked

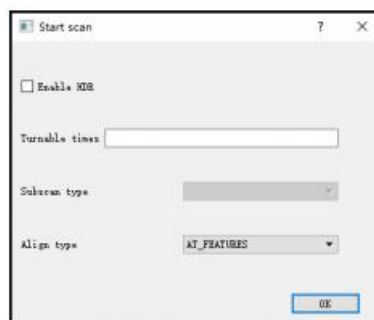
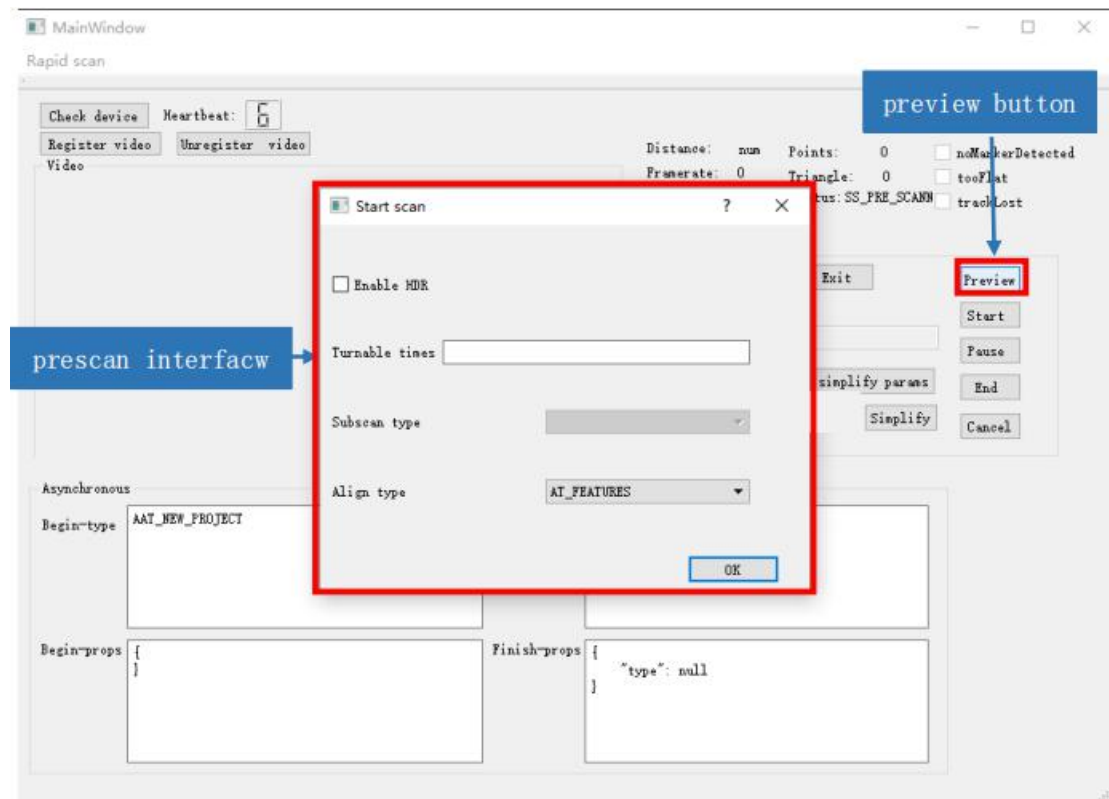


Figure 7 Preview interface

Table 3 Preview parameters

Parameter	Implication	UI Type	Function
Enable HDR	HDR enable	Check box	startscan.cpp->on_pushButton_clicked
Turnable times	Number of turns of the turntable	Text	startscan.cpp->on_pushButton_clicked
Subscan type	Scan subtype(gray, Rapid Scan has no subtype options)	Drop-down list	startscan.cpp->on_pushButton_clicked

Align type	Scan type: AT_FEATURES (Feature stitching) AT_MARKERS (Mark stitching) AT_HYBRID (Mixed stitching) AT_AUTO (Automatic stitching)	Drop-down list	startscan.cpp->on_pushButton_clicked
ok	Confirm entry to preview	Button	startscan.cpp->on_pushButton_clicked

Here our parameters are the default, click the "ok" button to enter the pre-scan mode. Click on "Register video" for camera video registration. If the previous operation is correct, the view of the scanner's left and right cameras can be displayed in real time under the video window, as shown in figure 8.

"Register video" Button: Camera Video Registration, mianwindow.cpp -> on_pushButton_RegisterProcessor_clicked

"UnRegister video" Button: Camera Video UnRegistration, mianwindow.cpp -> on_pushButton_UnRegisterProcessor_clicked

"Heartbeat" text: Server heartbeat (When the value is 0, the delegate server dies and can no longer provide data to the client). mianwindow.cpp->onHeartbeat

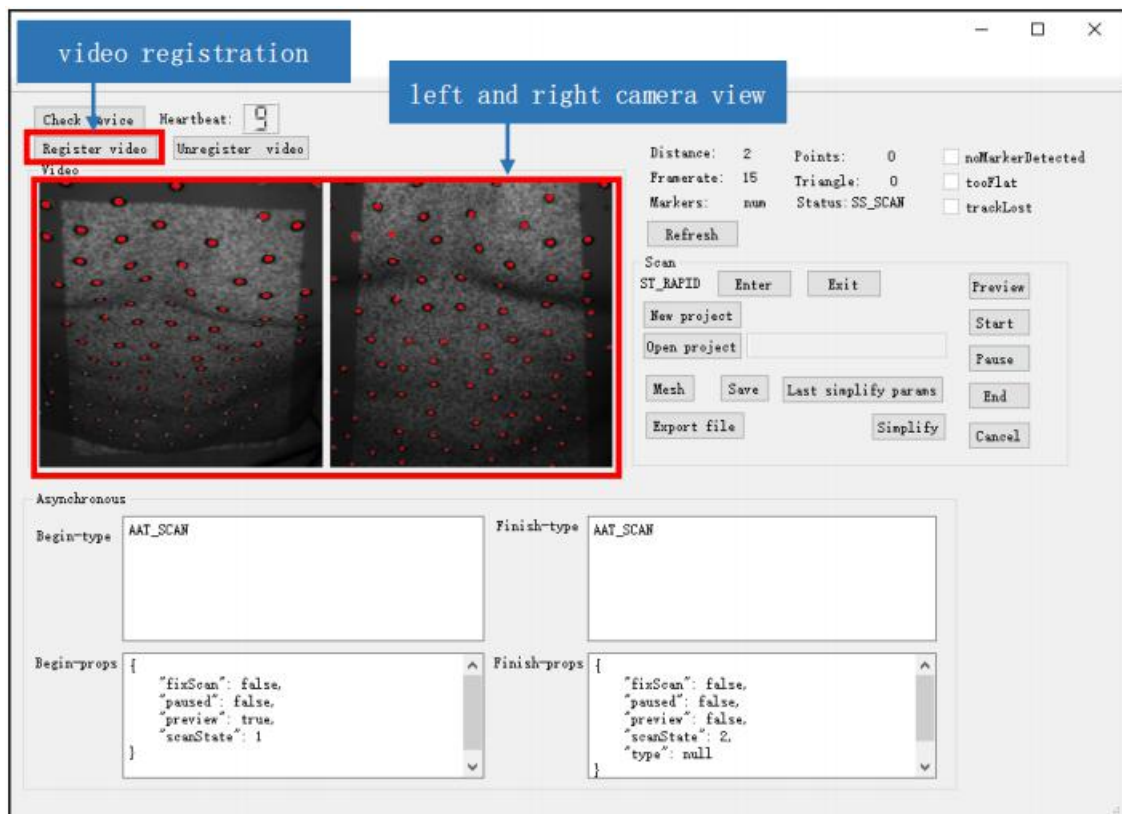


Figure 8 Sketch of Video Registration Interface

6、Start Scan

Click on the "Start" button to start scanning the model. The interface is shown in figure 9. The parameters of the start scan interface are the same as those of the pre-scan interface. Except the meaning of "ok" button is to enter the scan, the other parameters can be referred to in table 3. Here we use default parameters and click "ok" button to enter the scan .

Corresponding functions: mainwindow.cpp->on_pushButton_start_clicked

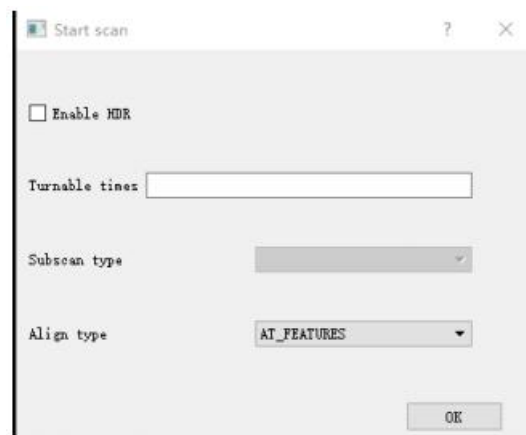
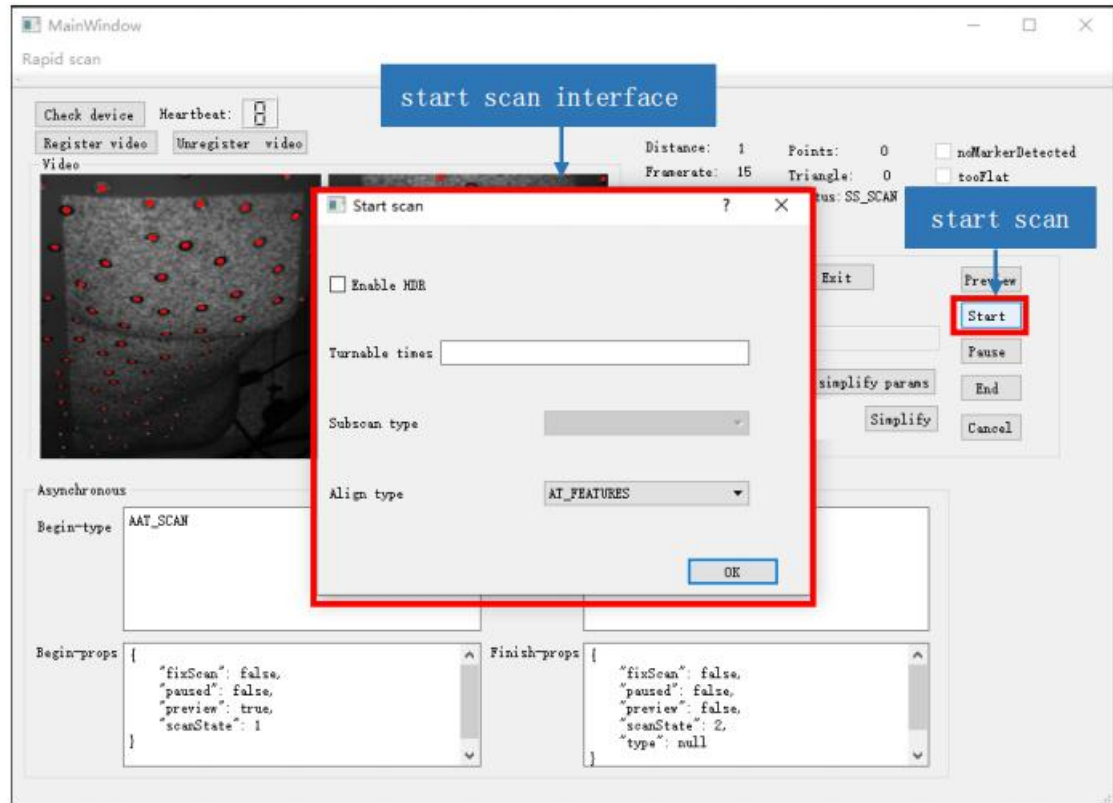


Figure 9 Start scan interface

As shown in figure 10, scan parameters such as “Heartbeat” , “Distance” , “ Points” , and “ Framerate” will be updated in real time during the scanning process. Clicking the Refresh button will also manually update the data.

“Refresh” Button: mainwindow.cpp->on_pushButton_Refresh_clicked

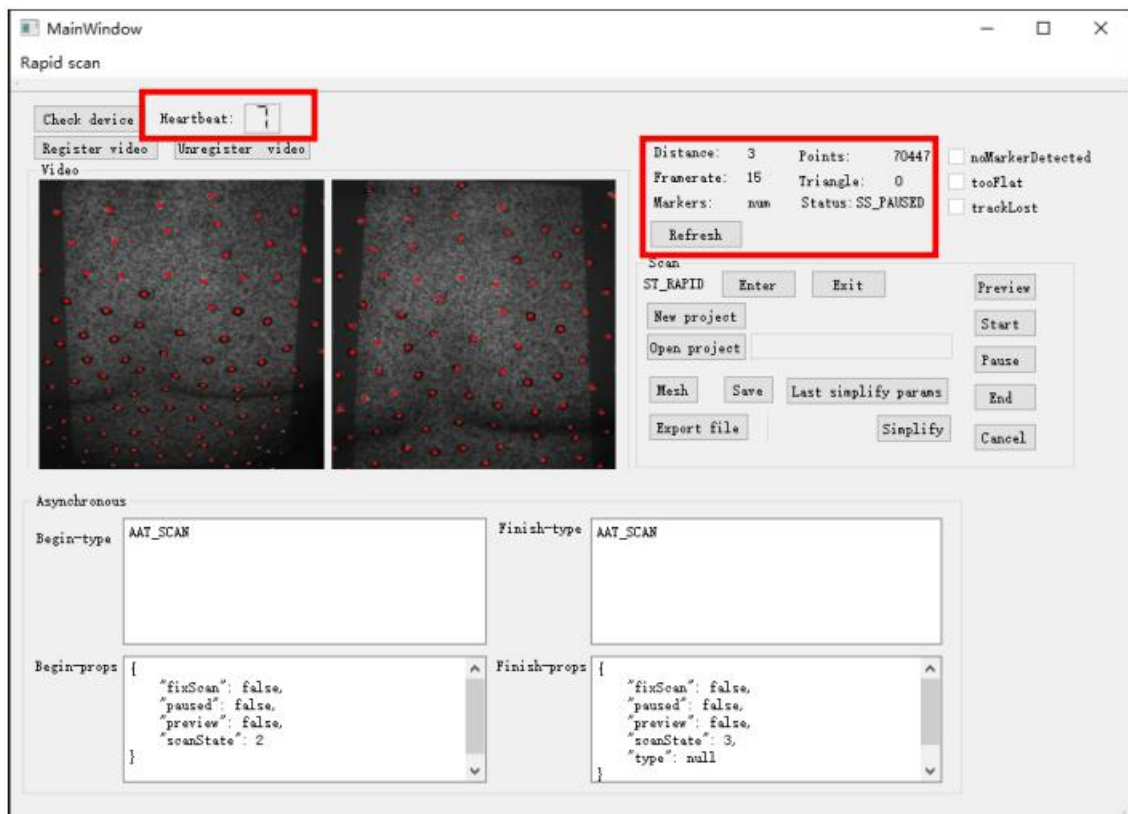


Figure 10 Schematic diagram of real-time data update during scanning

7、End Scan

When the data acquisition is completed, click the “End” button and set the parameters to end the scan, as shown in figure 11. The parameter details are shown in table 4. Here we are the default parameters, click the “ok” button to end the scan, and figure 12 is the end scan. A schematic diagram of a progress bar for post-data processing.

Corresponding functions: mainwindow.cpp->on_pushButton_scanEndScan_clicked

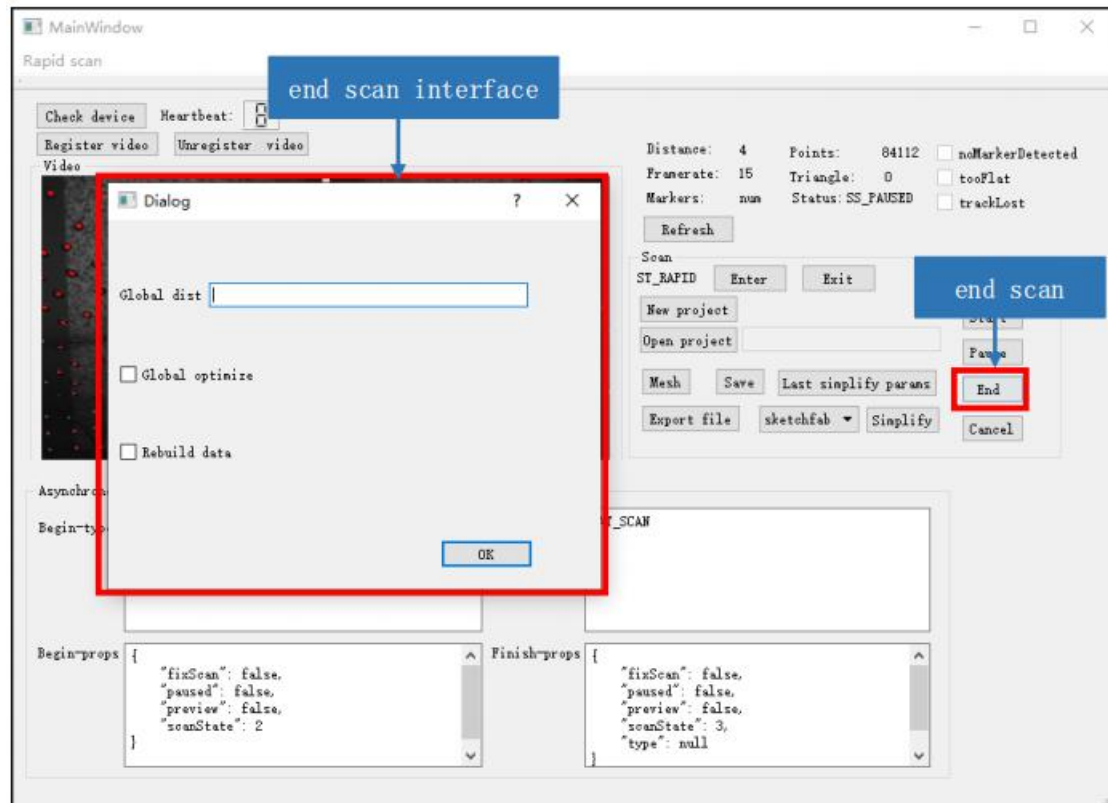


Figure 11 End scan interface

Table 4 End scan parameters

Parameter	Implication	UI Type	Function
Points dist	Point distance	Text	endscan.cpp-> on_pushButton_endScan_clicked
Global optimize	Global optimization	Check box	endscan.cpp-> on_pushButton_endScan_clicked
Rebuild data	Data reconstruction	Check box	endscan.cpp-> on_pushButton_endScan_clicked
ok	Confirm end scan	Button	endscan.cpp-> on_pushButton_endScan_clicked

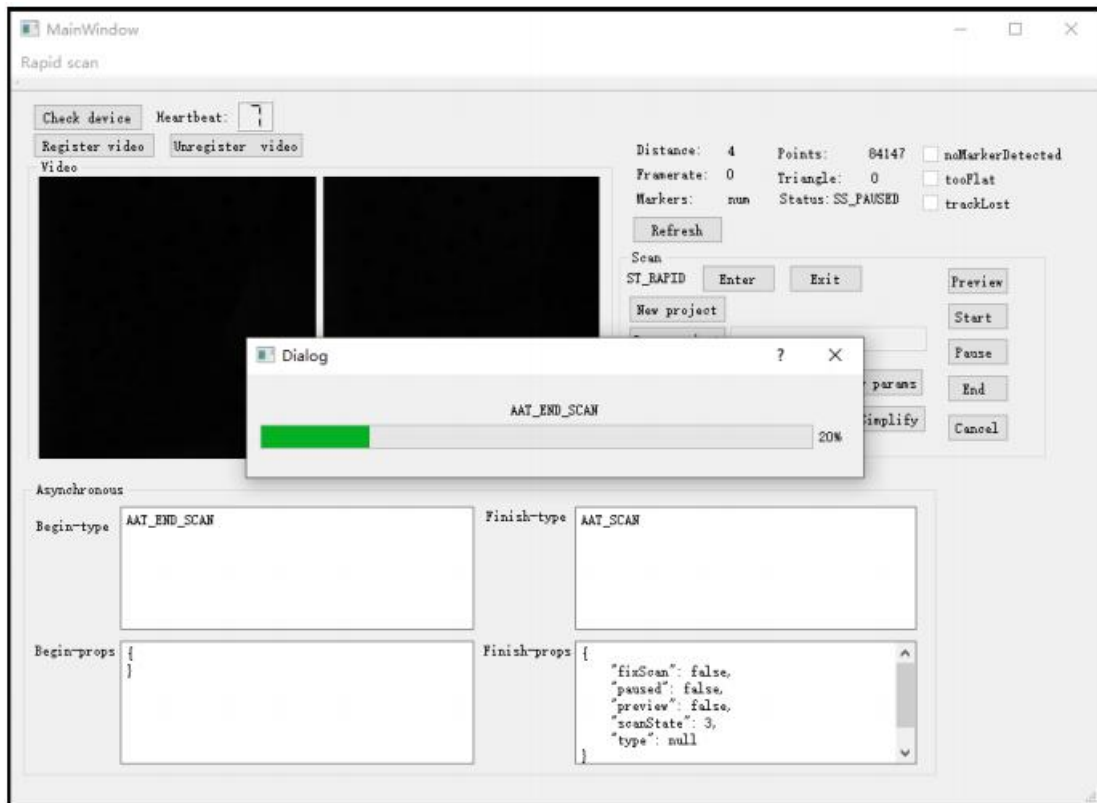


Figure 12 End scan data processing progress bar diagram

8、Mesh

Click the “Mesh” button to grid the point cloud. The schematic is shown in figure 13. The parameters of the Mesh processing interface are shown in table 5. We select the default parameters for grid processing.

“mesh” Button: mainWindow.cpp->on_pushButton_scanMesh_clicked

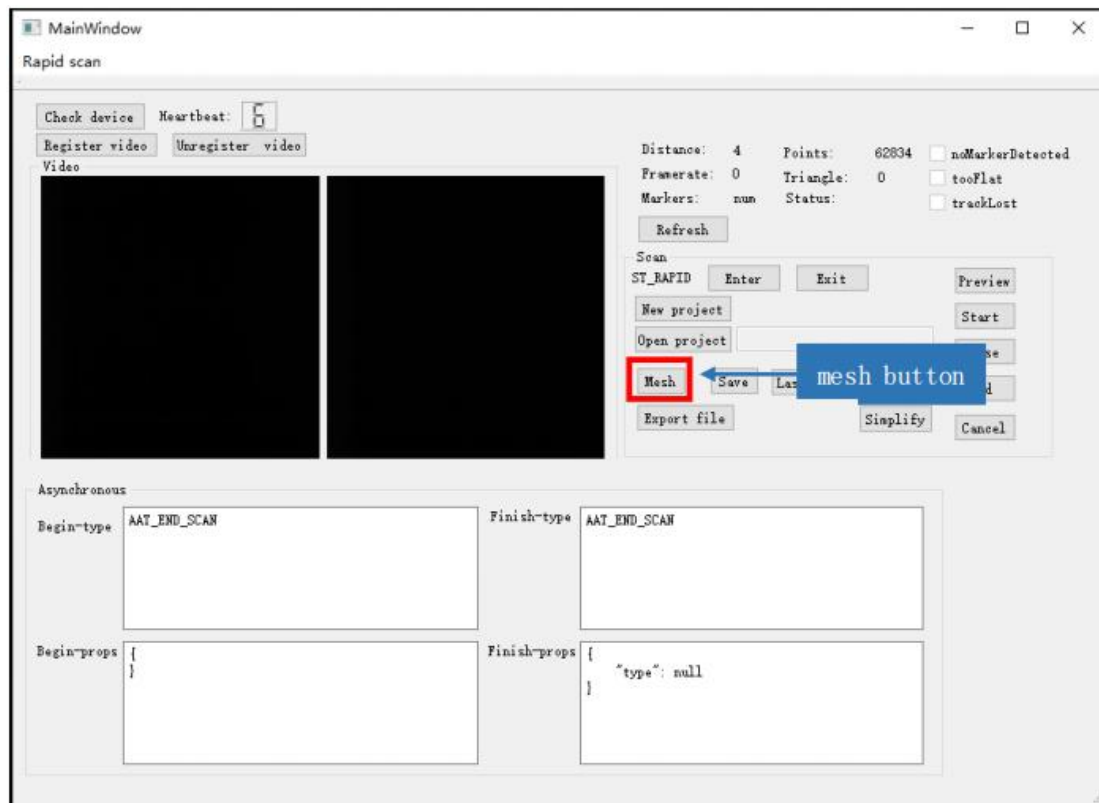


Figure 13 Mesh processing interface

Table 5 Mesh parameters

Parameter	Implication	UI Type	Function
Type	Grid encapsulation type: MT_NON_WATERTIGHT MT_WATERTIGHT	Drop-down list	mesh.cpp-> on_pushButton_mesh_clicked
Resolution	Mesh density: Hight Medium low	Drop-down list	mesh.cpp-> on_pushButton_mesh_clicked
ok	Confirm mesh process	Button	mesh.cpp-> on_pushButton_mesh_clicked

9、Simplify

Click the “Simplify” button to complete the simplification of the grid data. The simplified interface is shown in figure 14, the parameter details are shown in table 6. For details on the simplified parameters, check the “need mesh smooth” and “need mesh sharp” check boxes to simplify the mesh.

“Simplify” Button: mainwindow.cpp->on_pushButton_scanSimplify_clicked

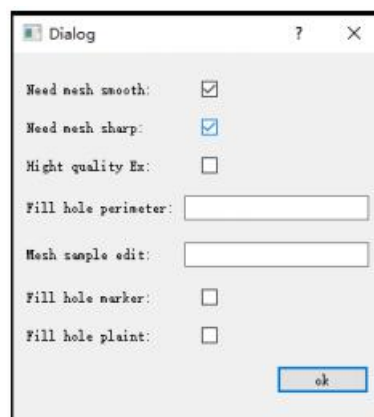
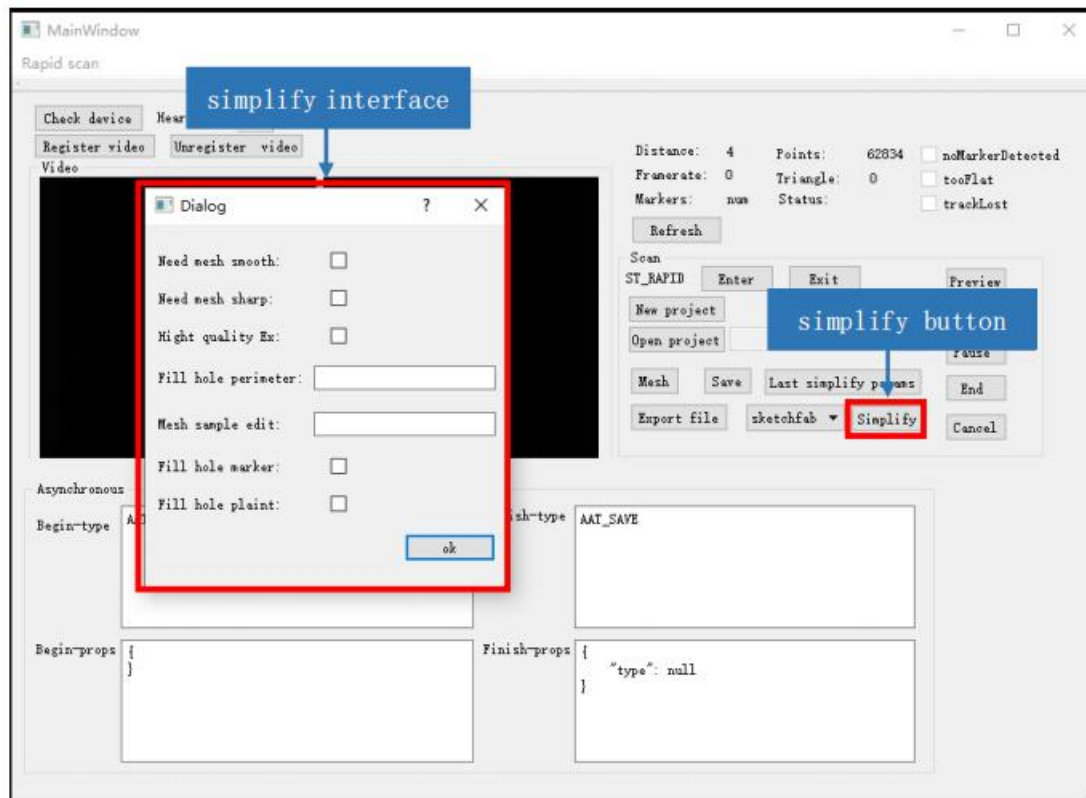


Figure 14 Simplify processing interface

Table 6 Simplify parameters

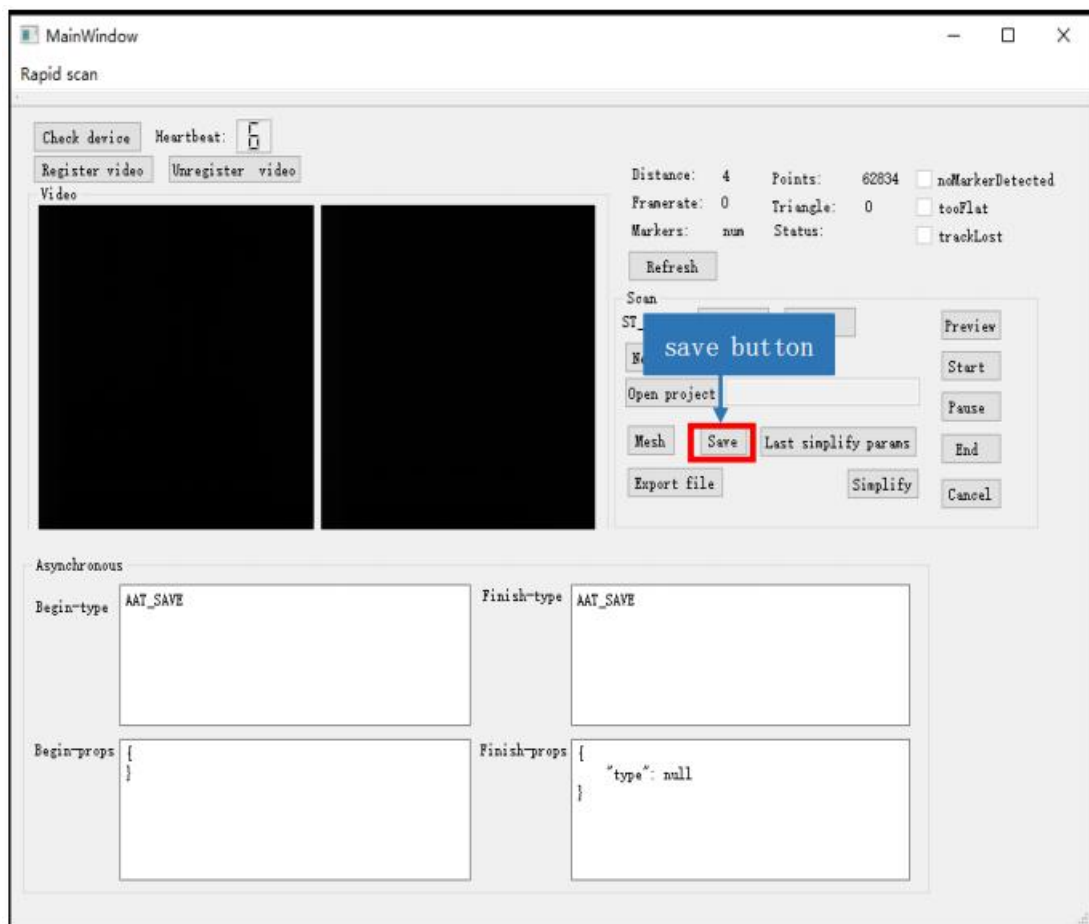
Parameter	Implication	UI Type	Function
Need mesh smooth	Mesh smoothing	Check box	simplify.cpp->on_pushButton_clicked
Need mesh sharp	Mesh sharp	Check box	simplify.cpp->on_pushButton_clicked
Hight quality Ex	high quality	Check box	simplify.cpp->on_pushButton_clicked
Fill hole perimenter	Fill hole parameter	Text	simplify.cpp->on_pushButton_clicked
Mesh sample edit	Simplified ratio	Text	simplify.cpp->on_pushButton_clicked

Fill hole marker	Fill hole marker	Check box	simplify.cpp->on_pushButton_clicked
Fill hole marker	Fill hole marker	Check box	simplify.cpp->on_pushButton_clicked
ok	Confirm entry into simplified operation	Button	simplify.cpp->on_pushButton_clicked

10、 Save

Click the “Save” button to save the model data. The interface is shown in figure 15. The details of the saved interface parameters are shown in table 7. Here we set the file save path, set the resize ratio to 100, and then click the “ok” button to save. (The size of the save scale does not affect the size of the file and will change the size of the model.)

“Save” Button :mainwindow.cpp->on_pushButton_scanSave_clicked



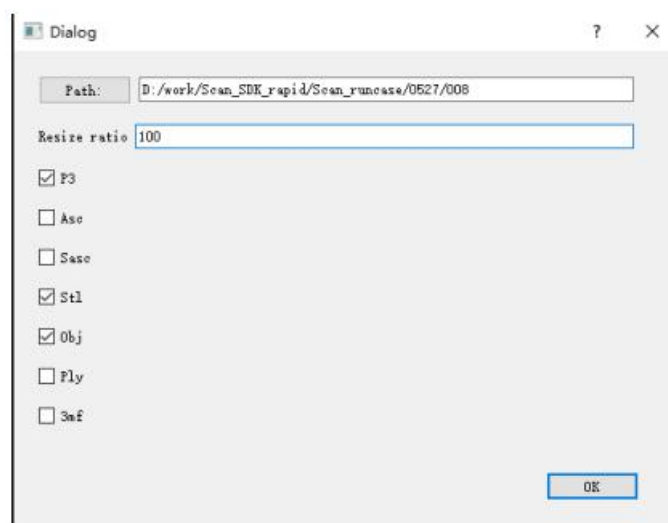
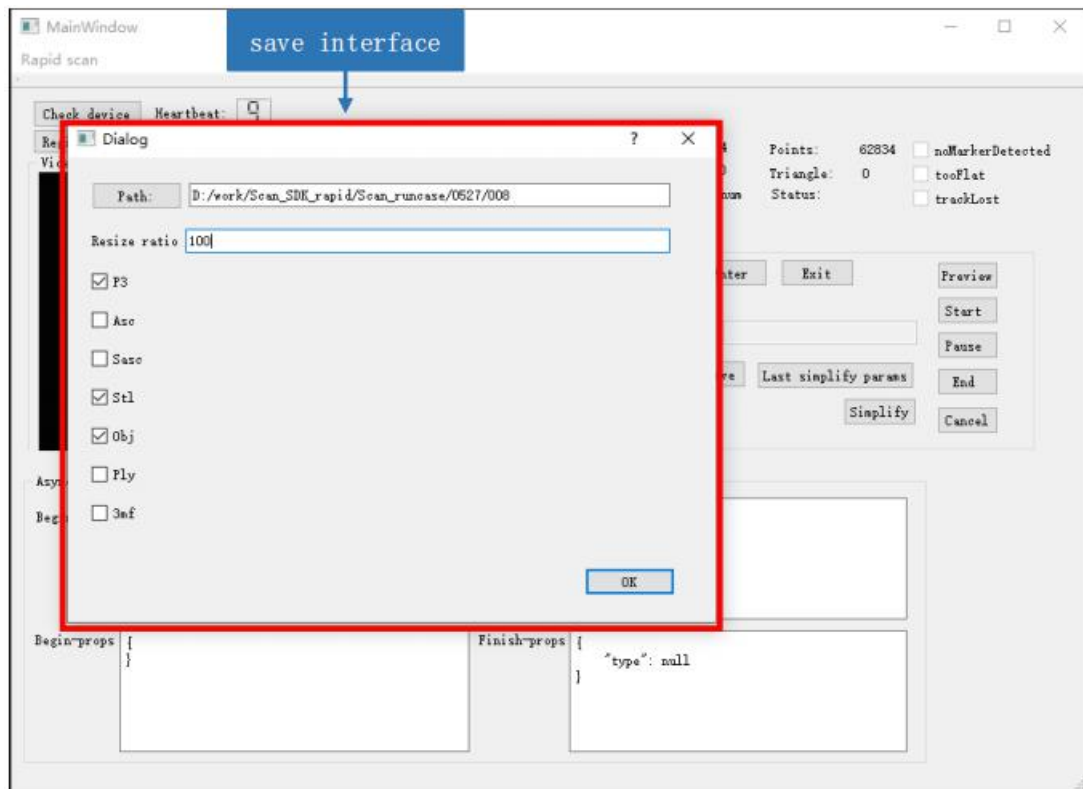


Figure 15 Save interface diagram

Table 7 Save processing parameters

Parameter	Implication	UI Type	Function
Path	Save route	Button+Text	save.cpp->on_pushButton_Path_clicked
P3、Asc、Sasc、Stl、Obj、Ply、3mf	File save format	Check box	save.cpp->on_pushButton_clicked
ok	Confirm the save process	Button	save.cpp->on_pushButton_clicked

11、The meaning of the parameters corresponding to the red box in figure 16, the parameter details are shown in table 8.

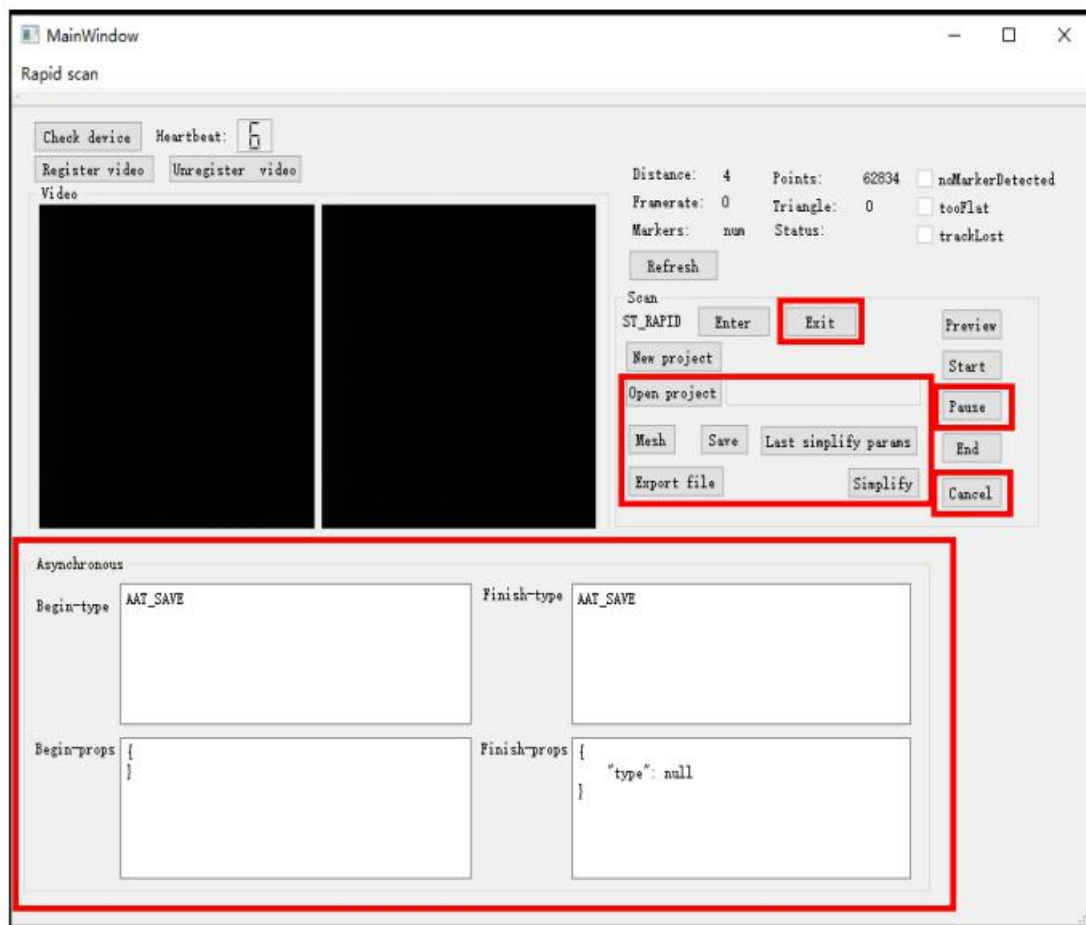


Figure 16 The selected parameter of the box

Table 8 Parameter list

Parameter	Implication	UI Type	Function
Exit	Exit the scan	Button	mainwindow.cpp-> on_pushButton_ScanExitScan_clicked
Open project	Open project(.rap_prj)	Button+Text	mainwindow.cpp-> on_pushButton_ ScanOpenProject_clicked
Pause	Pause scan	Button	mainwindow.cpp-> on_pushButton_Pause_clicked
Cancel	Cancel scan	Button	mainwindow.cpp-> on_pushButton_scanCancelScan_clicked
Mesh	Mesh package	Button	mainwindow.cpp-> on_pushButton_scanMesh_clicked
Save	Save project	Button	mainwindow.cpp-> on_pushButton_scanSave_clicked
Last simplify params	Get the latest simplified parameters	Button	mainwindow.cpp-> on_pushButton_ScanLastSimplifyParams_clicked

Export file	Export file	Button	mainwindow.cpp-> on_pushButton_ScanExportFile_clicked
Simplify	Simplify	Button	mainwindow.cpp-> on_pushButton_scanSimplify_clicked
Begin-type	Asynchronous callback begin type	Text	Mainwindow.cpp-> onPublishReceived
Begin-props	Asynchronous callback begin attribute	Text	mainwindow.cpp-> onPublishReceived
Finish-type	Asynchronous callback finish type	Text	mainwindow.cpp-> onPublishReceived
Finish-props	Asynchronous callback finish attribute	Text	mainwindow.cpp-> onPublishReceived

12、 After the scan is completed, three project files will be generated in the selected path. The suffixes are “.rap_prj”, “.dat”, and “.dat” respectively. The corresponding model file will be generated in the path of the saved model.

Attached:

1、 When the “Cancel” button is clicked, the interface is shown in figure 17, the parameter details are shown in table 9.

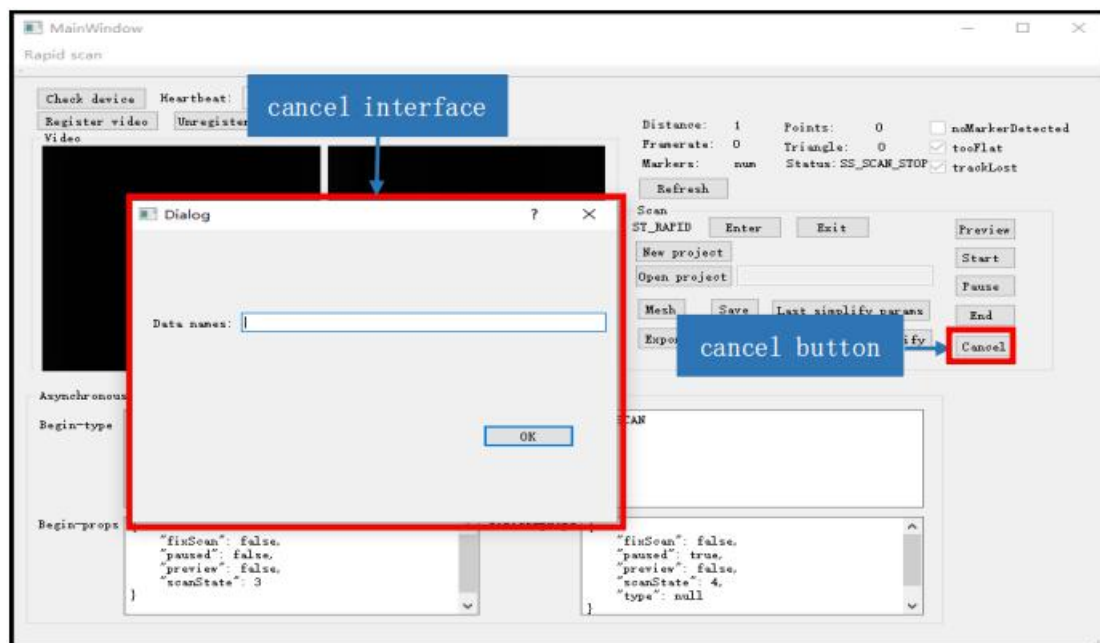


Figure 17 Cancel the scanning interface

Table 9 Cancel scan parameter table

Parameter	Implication	UI Type	Function
Data names	Canceled file name	Text	cancel.cpp->on_pushButton_cancelScan_clicked
ok	Confirm cancellation	Button	cancel.cpp->on_pushButton_cancelScan_clicked

2、Click the "Pause" button to pause the scan, as shown in figure 18, the interface is consistent with figure 9. When the scan is paused and then the scan is continued, the "Pause" button can be clicked here, and when the "End" button is clicked, the scan ends.

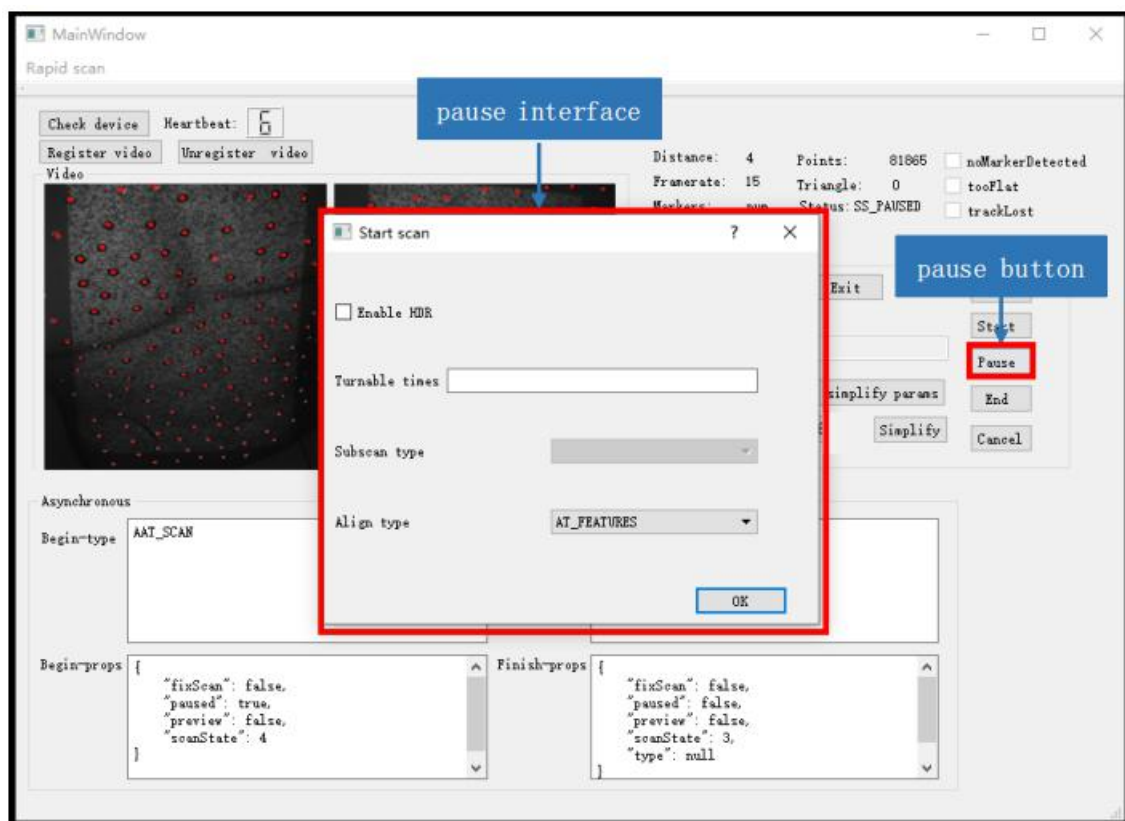


Figure 18 Pause the scanning interface

3、Click on the "Exit" button to exit the ongoing scan project. Rescan can start a new project scan from step 3 in the document.