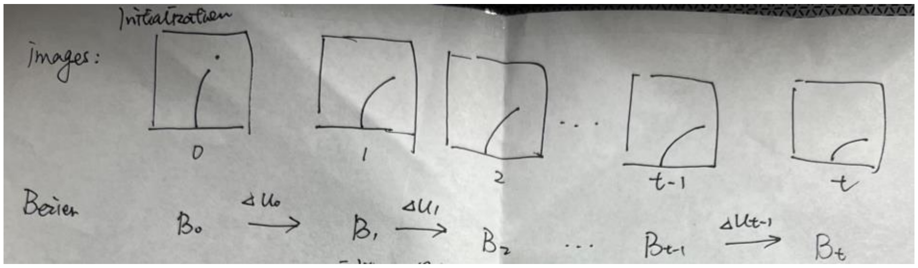
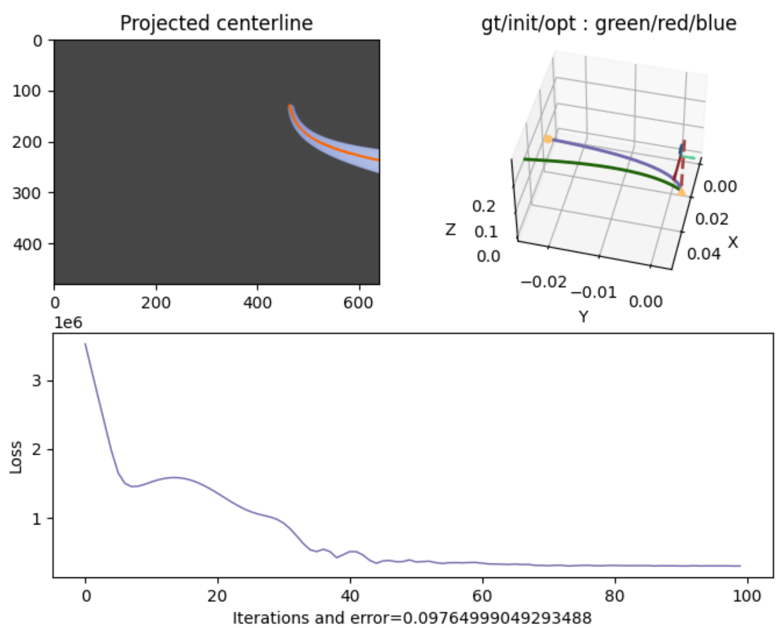
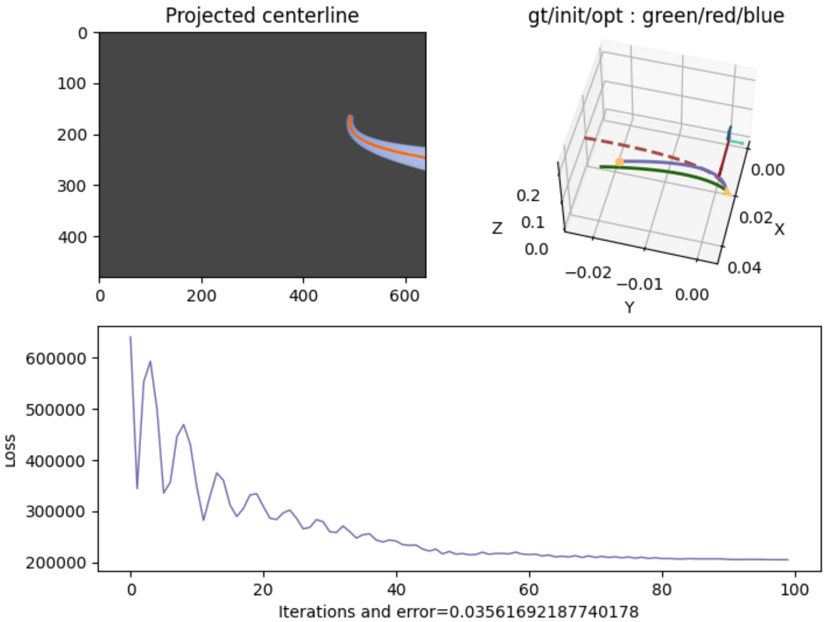


Method 1:

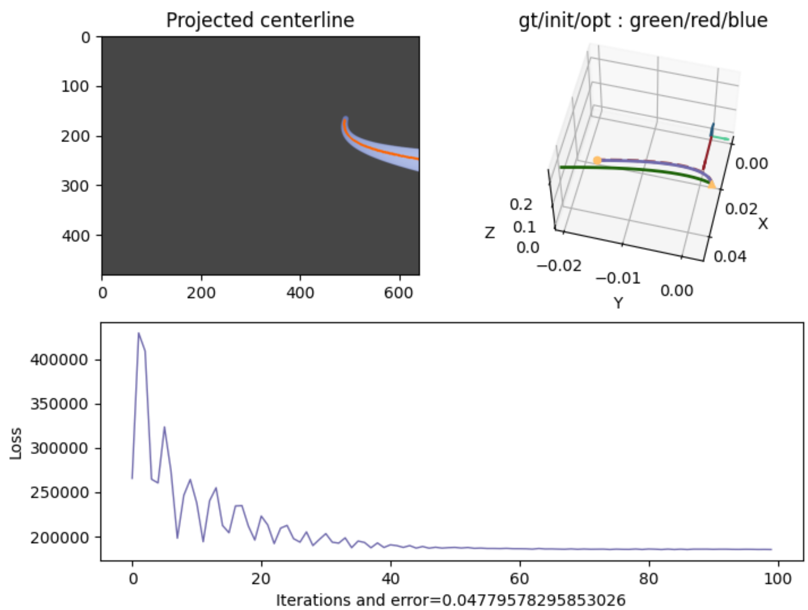
① $\arg\min_{B_t} \sum_{i=1}^{t} w_i \mathcal{L}_I\left(h^{-1}\left(h(B_t) - \sum_{k=1}^{i} \Delta u_{t-k}\right), I_{t-i}\right)$ Method 1

$$B_{t-i}^{(0)} = h^{-1}\left(h(B_t^{(0)}) - \sum_{i=0}^{i} \Delta u_{t-k}\right) \Leftarrow$$

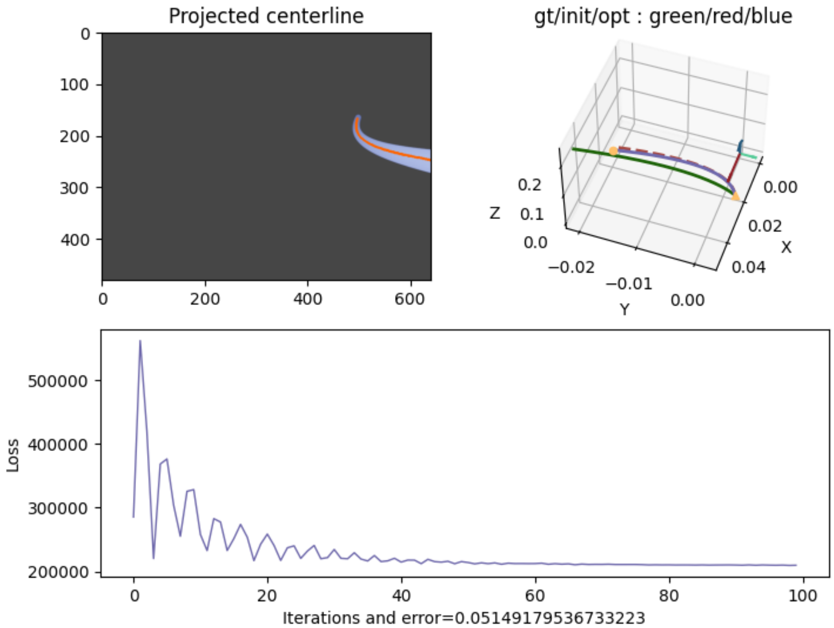




Projected centerline
gt/init/opt : green/red/blue
Z
X
Y
Loss
Iterations and error=0.03561692187740178



Projected centerline
gt/init/opt : green/red/blue
Z
X
Y
Loss
Iterations and error=0.04779578295853026



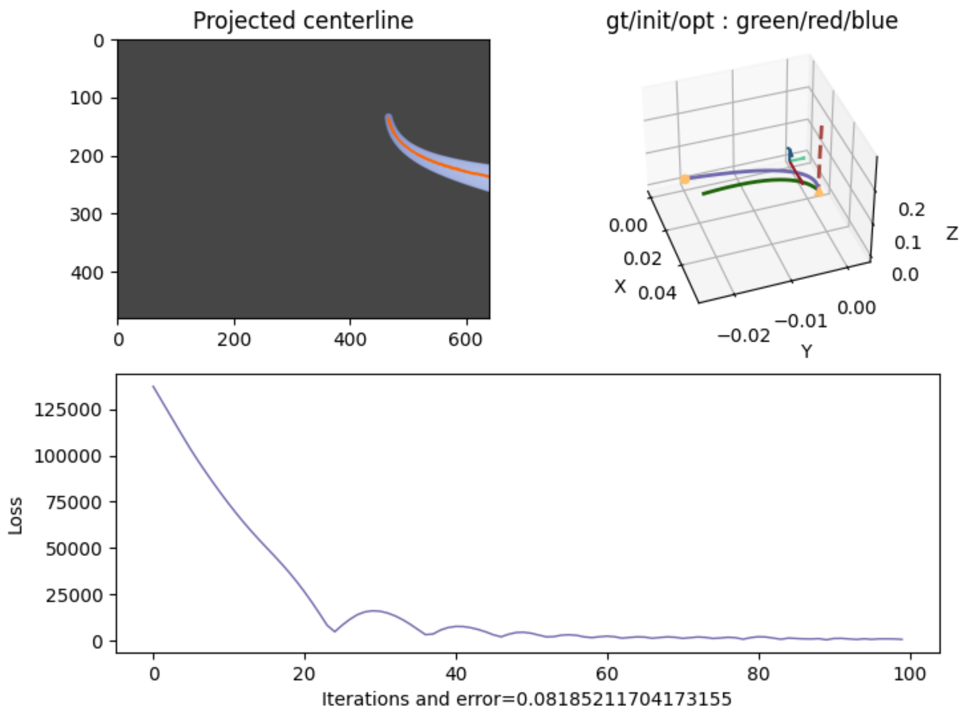
We can see that in the reconstruction result of method one, as more photos are used, the error will accumulate.

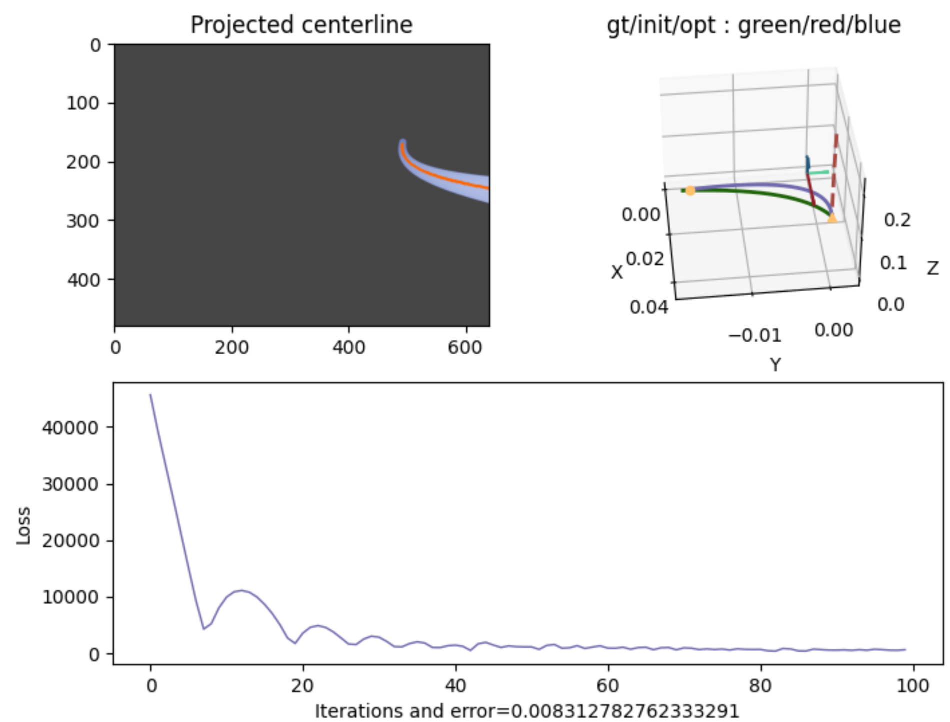
Method 2:

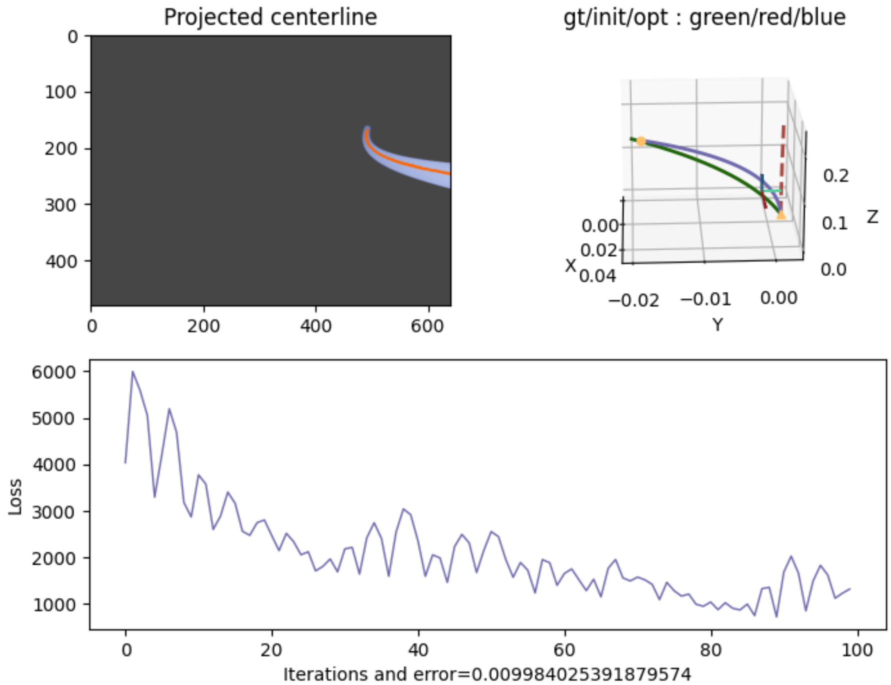
③ $\underset{B_0, B_1, \ldots B_t}{\operatorname{argmin}} \sum_{i=1}^{t} \omega_{1,i} \mathcal{L}_I(B_{t-i}, I_{t-i}) + \sum_{i=0}^{t-1} \overset{\omega_{2,i}}{} \mathcal{L}_B(h^{-1}(h(B_{t-i}) - \Delta U_{t-i-1}), B_{t-i-1})$

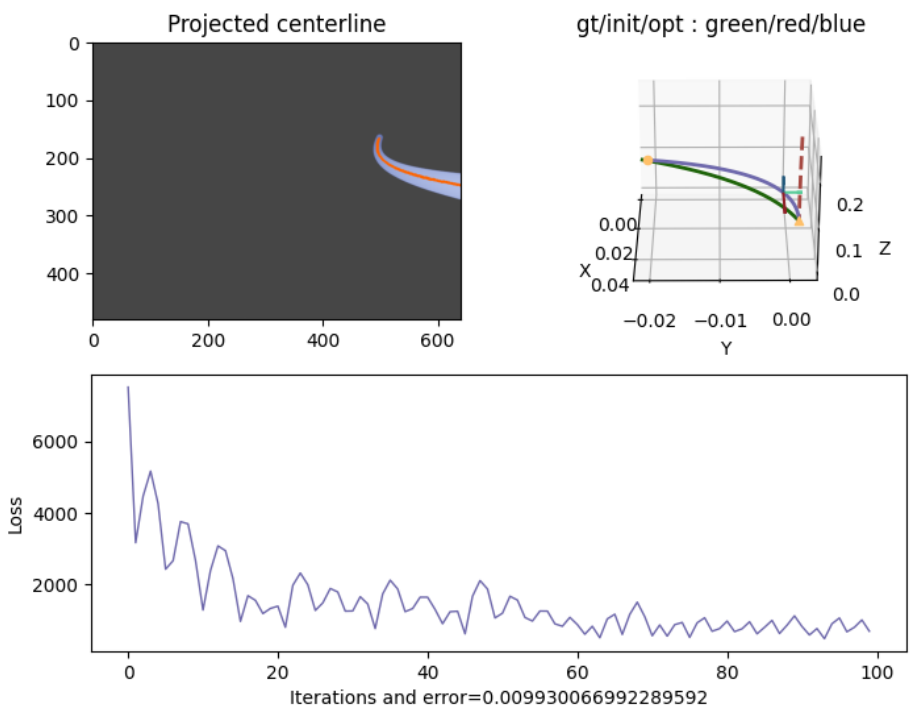
straight line.

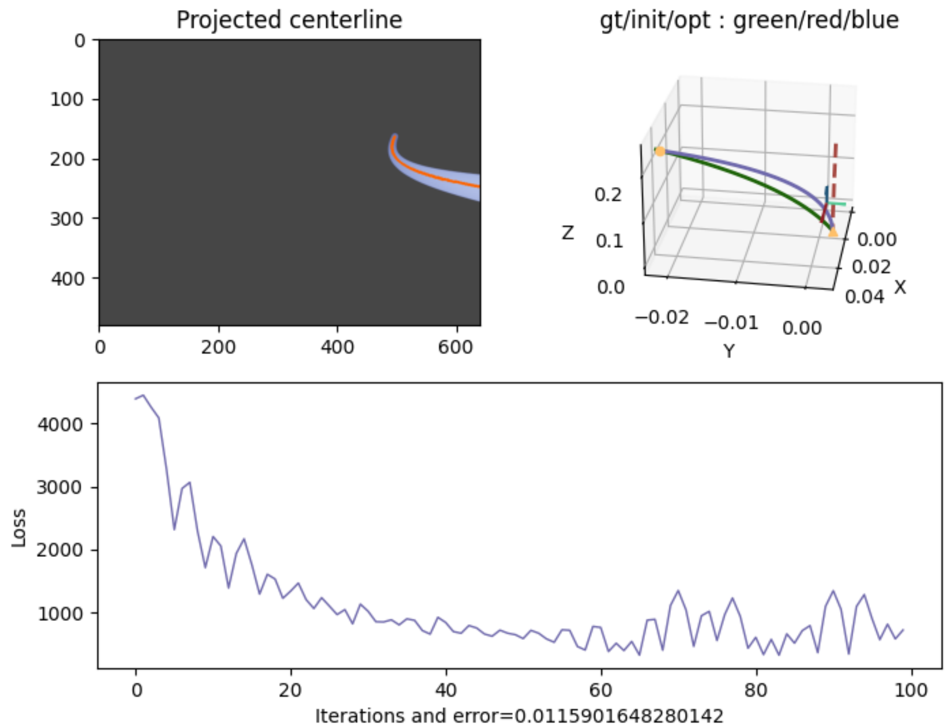
Constantly being updated


Projected centerline
gt/init/opt : green/red/blue
0
100
200
300
400
0
200
400
600
0.00
0.02
X
0.04
−0.02
−0.01
0.00
Y
0.2
0.1
0.0
Z
125000
100000
75000
50000
25000
0
Loss
0
20
40
60
80
100
Iterations and error=0.08185211704173155


Projected centerline
gt/init/opt : green/red/blue
0
100
200
300
400
0
200
400
600
0.00
0.02
X
0.04
−0.01
0.00
Y
0.2
0.1
0.0
Z
40000
30000
20000
10000
0
Loss
0
20
40
60
80
100
Iterations and error=0.008312782762333291


Projected centerline
gt/init/opt : green/red/blue
X
Y
Z
Loss
Iterations and error=0.009984025391879574


Projected centerline
gt/init/opt : green/red/blue
X
Y
Z
Loss
Iterations and error=0.009930066992289592



The error matrix:

```
[[0.08185212 0.         0.         0.         0.        ]
 [0.08406667 0.00831278 0.         0.         0.        ]
 [0.08973485 0.01059425 0.00998403 0.         0.        ]
 [0.08439061 0.00850765 0.01101244 0.00993007 0.        ]
 [0.08369952 0.00961126 0.01294547 0.01004472 0.01159016]]
```

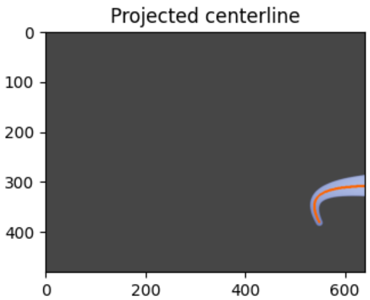
Every row represents the number of learning steps

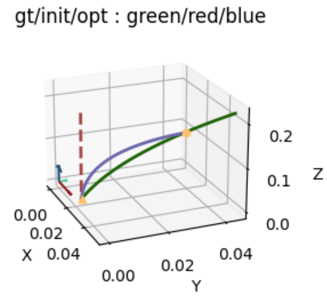
Every column represents a certain frame of catheter

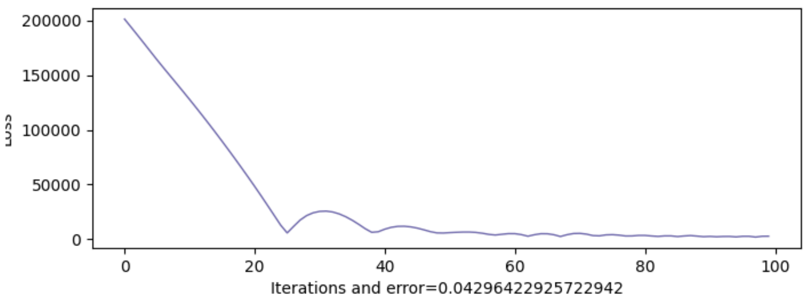
We can see that using this method, the catheter can be reconstructed more accurately. The tip position reconstructed fits the ground truth very well with two or more frames applied. However, when the learning step increases, the time used to converge increases greatly.

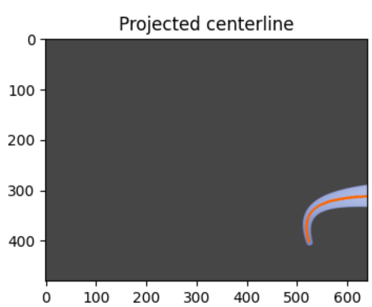
Below are other cases:

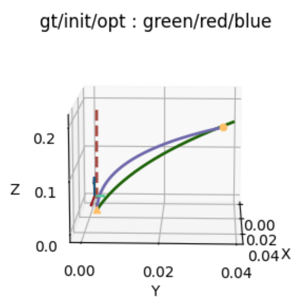
Case 1

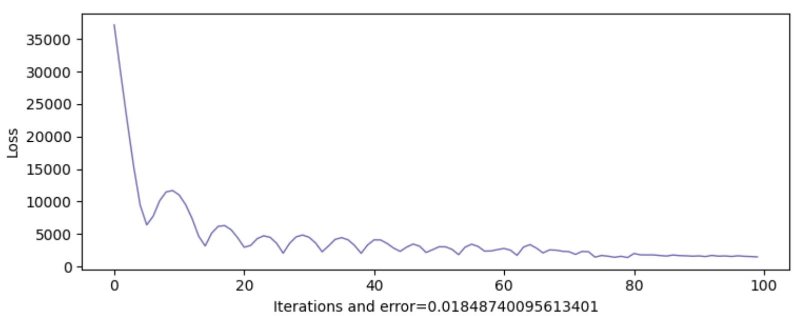


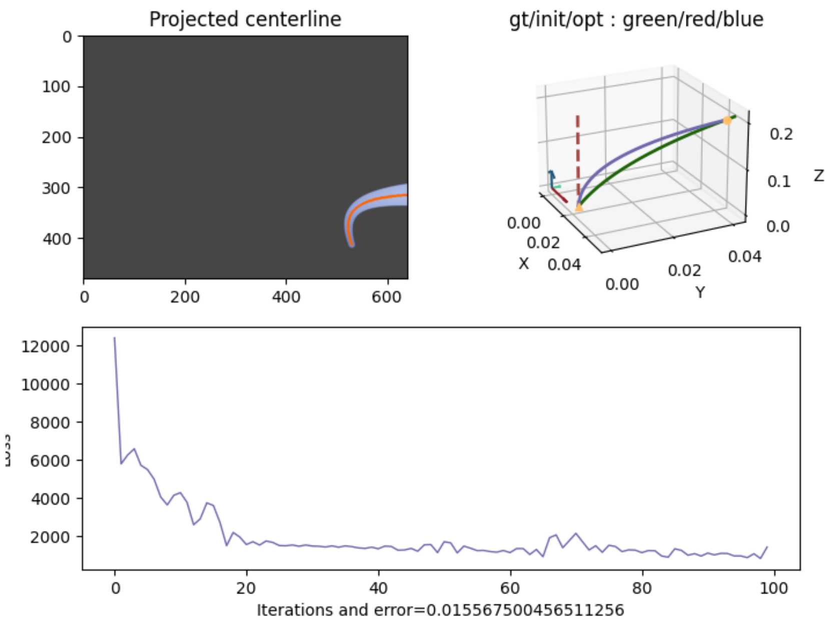


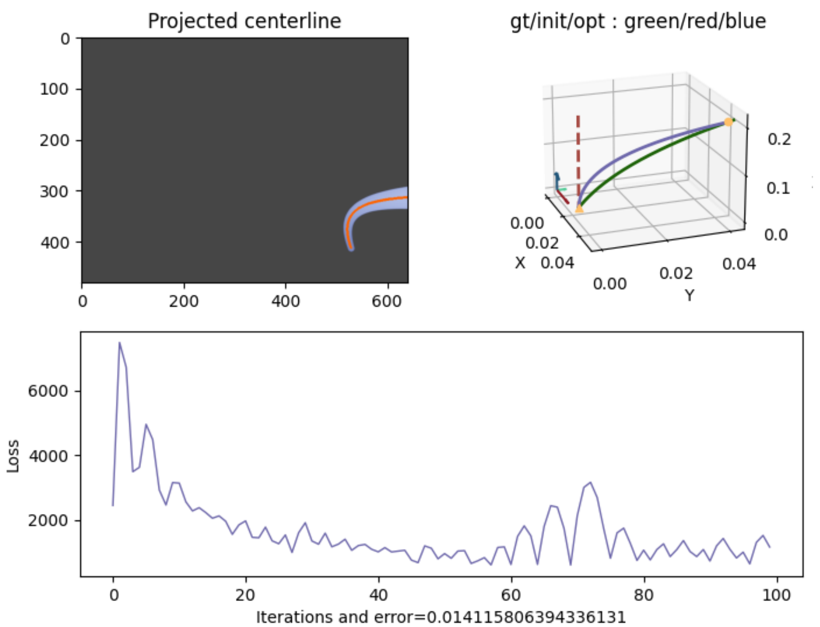


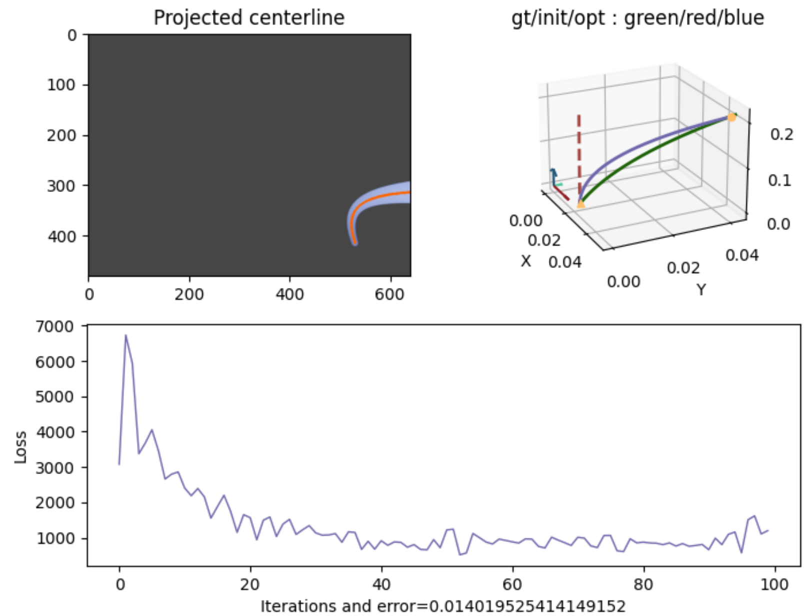




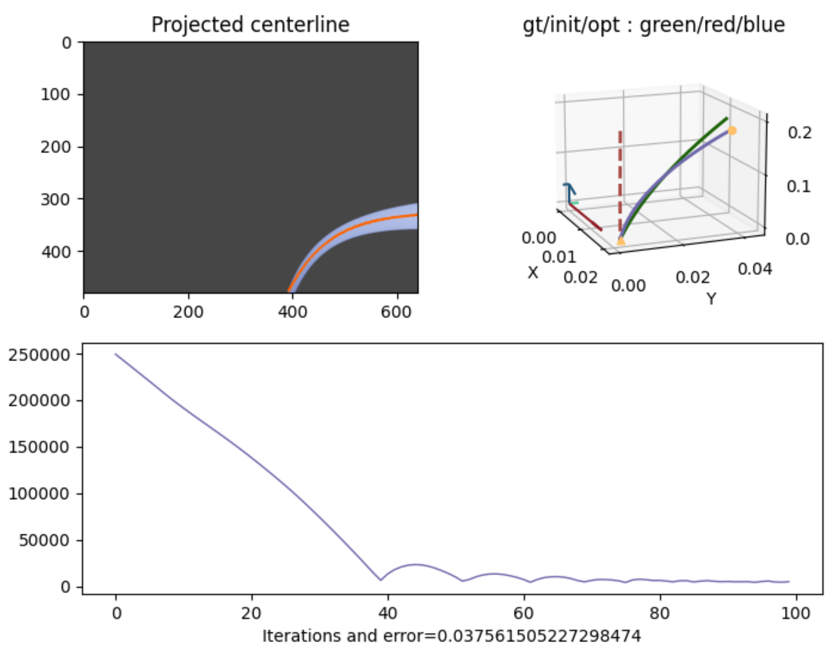


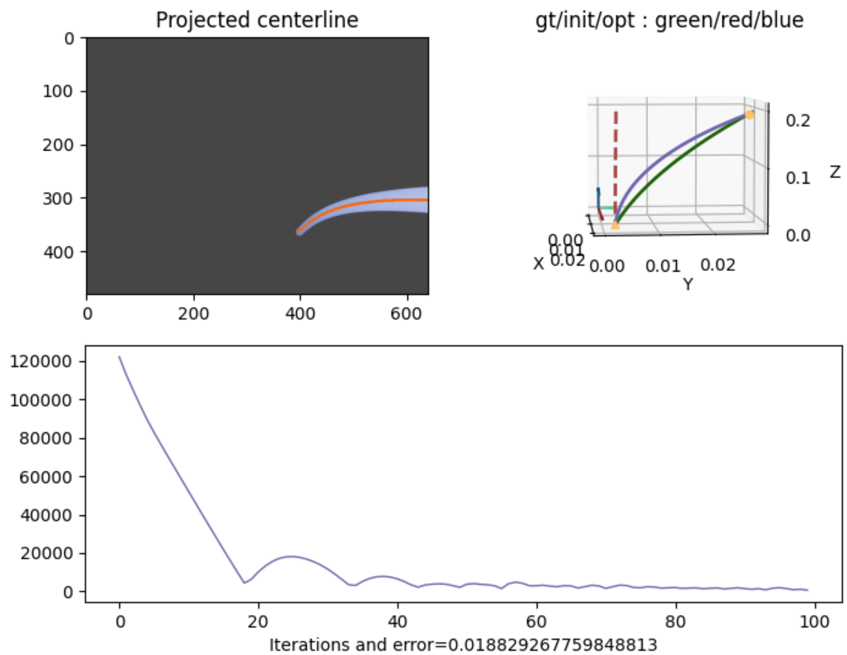

Projected centerline
gt/init/opt : green/red/blue
X
Y
Z
Iterations and error=0.015567500456511256

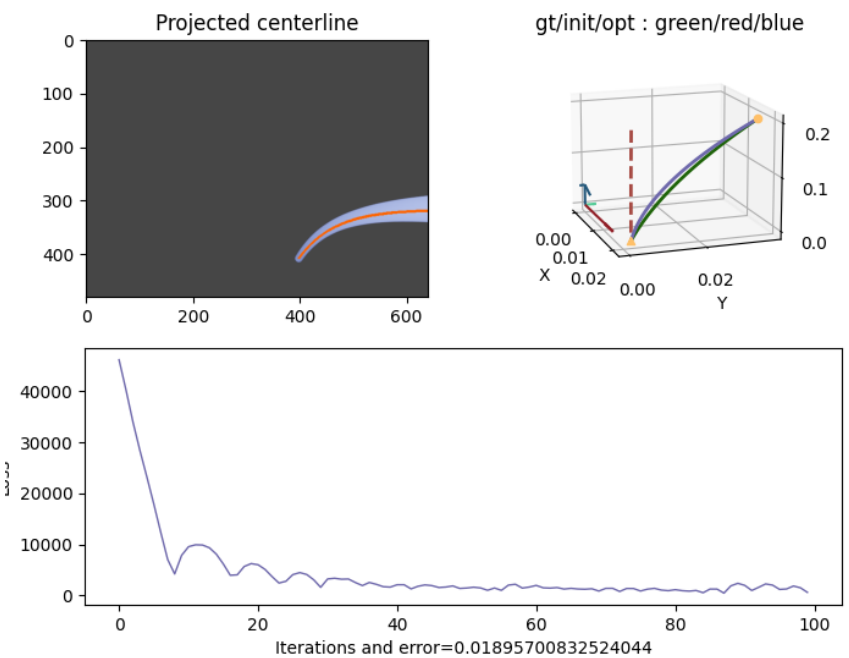

Projected centerline
gt/init/opt : green/red/blue
X
Y
Loss
Iterations and error=0.014115806394336131

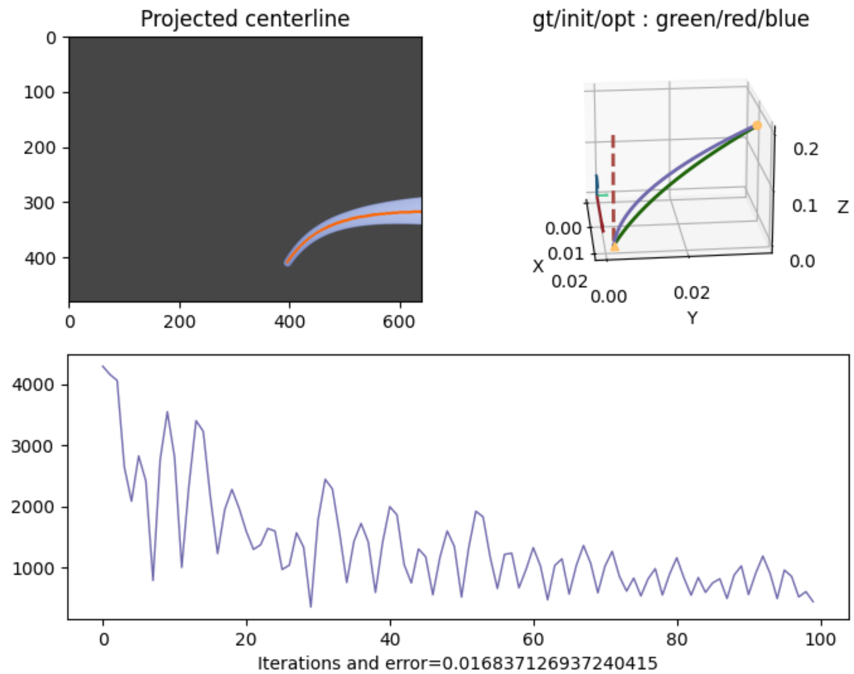

Projected centerline
gt/init/opt : green/red/blue
X
Y
Loss
Iterations and error=0.014019525414149152

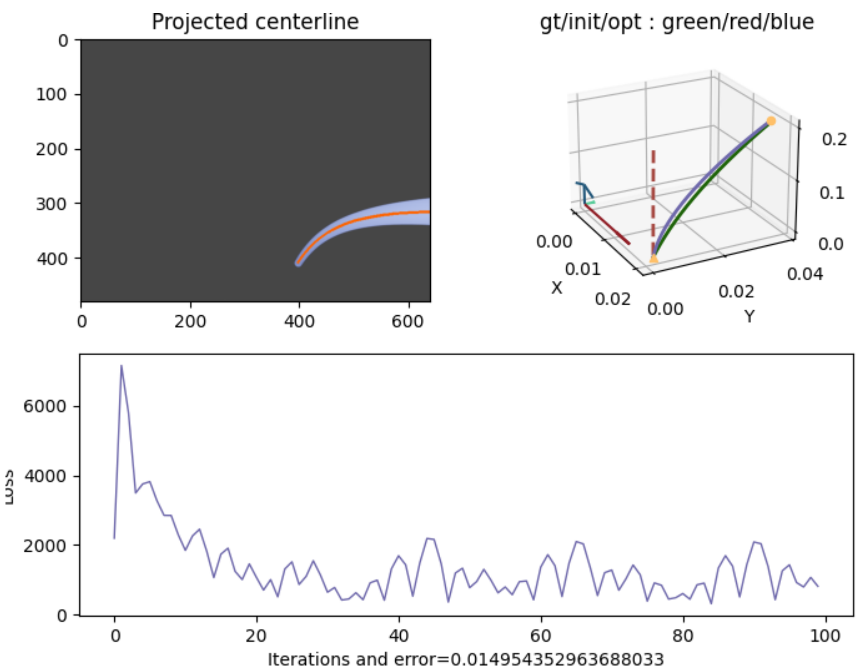
Case 2:


Projected centerline
gt/init/opt : green/red/blue
X
Y
Iterations and error=0.037561505227298474


Projected centerline
gt/init/opt : green/red/blue
X
Y
Z
Iterations and error=0.018829267759848813


Projected centerline
gt/init/opt : green/red/blue
X
Y
Iterations and error=0.01895700832524044






```
[[0.03756151 0.         0.         0.         0.        ]
 [0.03264926 0.01882927 0.         0.         0.        ]
 [0.04007884 0.02788116 0.01895701 0.         0.        ]
 [0.04542703 0.03305898 0.02390212 0.01683713 0.        ]
 [0.05128393 0.03894542 0.02988043 0.02233291 0.01495435]]
```