Быстрый запуск OpenVO.

1. Для использования алгоритма визуальной одометрии, скачайте пакет Python “Open\_VO” с репозитория на гитхабе используя “python/build/Open\_VO.cp312-win\_amd64.pyd” или “python/build/open\_vo-1.0.0-py3-none-any.whl”.
2. Создайте проект в PyCharm (или другой IDE) и создайте \*.py файл. Установите “Open\_VO”:
   1. “.pyd”: положите пакет “Open\_VO.cp312-win\_amd64.pyd” в ваш проект и импортируйте библиотеку “import Open\_VO”.
   2. “.whl”: установите пакет “pip install python/build/open\_vo-1.0.0-py3-none-any.whl” и импортируйте библиотеку “import Open\_VO”.

3. Импортируйте следующие модули:

import cv2  
import time  
import math # For math.nan if needed  
from ovo\_types import CameraParams, Pos\_i2  
from video\_processor\_ovo import VideoProcessorOVO  
from ovo\_constants import OVO\_ANGLES\_FROM\_VIDEO # or OVO\_ANGLES\_FROM\_SOURCE

\*\*Модули используют OpenCV бэкенд для работы с видеофайлами / видеопотоками, ORB и визуализацией видео в примерах.

4. После чего используйте следующую конструкцию:

4.1. Опишите параметры:

def main():  
 cam\_p = CameraParams()  
 cam\_p.fov = 86 # Assuming 86 was degrees, convert to radians if map\_scale expects it  
 # Check how map\_scale\_py uses fov. If it's directly in tan(fov), it should be radians.  
 # If fov in C++ Camera\_params was degrees, then the C++ mapScale would convert.  
 # Python's math.tan expects radians.  
  
 res = Pos\_i2(1920, 1280)  
 cam\_p.resolution = res  
 cam\_p.type = 0 # Example type  
  
 video\_file = "./examples/test.LRF" # Make sure this file exists

4.2. Инициализируйте класс для работы с видео “VideoProcessOVO”:

vpo = VideoProcessorOVO(  
 p\_params=cam\_p,  
 source\_info=video\_file,  
 api\_reference\_or\_source\_flag=OVO\_ANGLES\_FROM\_VIDEO, # SOURCE\_FLAG  
 custom\_shape\_or\_max\_points=100 # maxPoints

except IOError as e:  
 print(f"Error initializing VideoProcessor: {e}")  
 return

vpo.set\_custom\_shape(640, 640)  
h\_altitude = 300.0 # meters  
k = 0 # key code

4.3. Получите кадр и отправьте телеметрию:

while k != 27: # 27 is ASCII for ESC  
 start\_time = time.perf\_counter()  
  
 vpo.set\_data\_for\_one\_iteration(h\_altitude) # Only setting altitude  
  
 if vpo.grab\_frame\_and\_data():  
 end\_time = time.perf\_counter()  
 processing\_time\_ms = (end\_time - start\_time) \* 1000  
 print(f"Full tick: {processing\_time\_ms:.2f} ms")  
  
 pos = vpo.trajectory.get\_curr\_pos()  
 print(f"Position x,y,z: {pos.x:.2f}, {pos.y:.2f}, {pos.z:.2f}")  
  
 current\_display\_frame = vpo.get\_frame() # This is the processed gray frame  
 if current\_display\_frame is not None:  
 cv2.imshow("Capture OVO Python", vpo.frame)  
 else:  
 print("No frame to display from VPO.")  
 else:  
 # print("grab\_frame\_and\_data failed or no new data.")  
 # If video ends, cap.read() will return False, bgr\_frame will be None  
 # Check if cap is still good  
 if not vpo.cap.isOpened() or vpo.cap.read()[0] == False : # Heuristic check if stream ended  
 print("Video stream ended or cannot read frame.")  
 break  
  
  
 k = cv2.waitKey(1) & 0xFF # Wait for 1 ms

5. Освободите память:

cv2.destroyAllWindows()  
 if vpo.cap.isOpened():  
 vpo.cap.release()  
 print("Processing finished.")  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

6. Для запуска нажмите Run (Shift + F10).

7. Вывод консоли:

Position x,y,z: -277.98, 280.68, 300.00

Full tick: 12.98 ms

Position x,y,z: -282.94, 284.75, 300.00

Full tick: 12.48 ms

Position x,y,z: -287.45, 288.81, 300.00

Full tick: 12.37 ms

Position x,y,z: -291.97, 292.87, 300.00

Full tick: 13.50 ms

Таким образом, это простой пример использования алгоритм визуальной одометрии. «Position x,y,z: -291.97, 292.87, 300.00» - информация о позиции в локальной системе координат. Zero Point – точка старта видеопотока. Переменные x,y,z представлены в метрах. При необходимости поворота угла оси Oy, используйте affine\_params.angle из VideoProcessorOVO (video\_processor\_ovo.py).