

第七组激光雷达方舱实验报告

宋熙喆 PB21000301

1 实验背景

米散射是从气溶胶粒子中弹性散射的激光辐射，米散射激光雷达即利用大气气溶胶的米散射来反演大气气溶胶信息的激光雷达，借此获得大气消光系数，从而计算大气气溶胶的密度分布。

激光雷达方舱承载全套激光雷达系统。本次实验，我们第七组于激光雷达方舱中操作米散射雷达并分析原始回波信号数据，计算大气剖面的消光系数。

2 实验目标

开启激光雷达并记录三十分钟的回波数据，反演大气消光系数。

3 实验软硬件调试部分

1. 打开激光器电源。O 为关闭，I 为打开。
2. 打开激光器显示面板。钥匙转至 0 时为关，转至 1 为开。
3. 激光器初始化。
4. 点击激光标志，打开激光器预热。
5. 在信号采集器设置采集点数，信号类型，采样率。
6. 打开信号采集器，设置采集组数和脉冲积分时间。

如图，在采集前安装后继光路，即安装准直透镜、滤光片、汇聚透镜。一端的银色传输光纤连向望远镜，另一端则连向电脑。

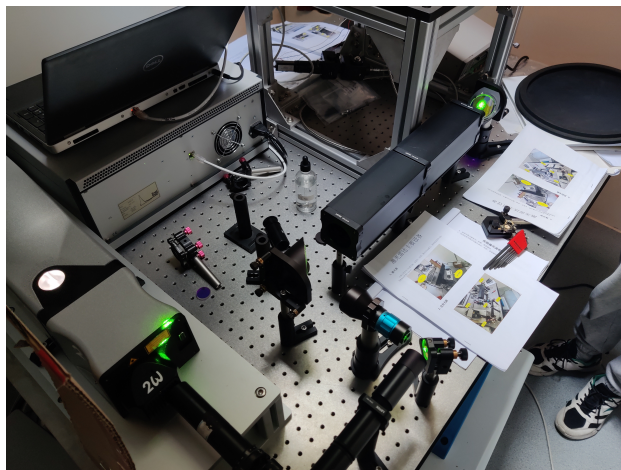


图 1: 光路图

3.1 光路调试部分

完成激光雷达安装，必须确保激光信号与望远镜轴大致平行，使望远镜接收到足够强的激光后向散射信号。

安装激光出射光路。最后一个反射镜与光学平台大致呈 45° 角，使得激光光路垂直于光学平台并射入大气。出射激光在照射到玻璃窗会反射而形成绿色光点。粗调反射镜的 2 个旋钮以调整反射镜面方向，直到两者大致平行。

细调反射镜 2 旋钮，使得信号采集器界面在开始信号采集后出现非 0 数据。

3.2 数据采集部分

在完成准直后，设置信号采集器接收三十分钟长的信号。采集时间为 2024 年 5 月 28 日 18:17-18:46。

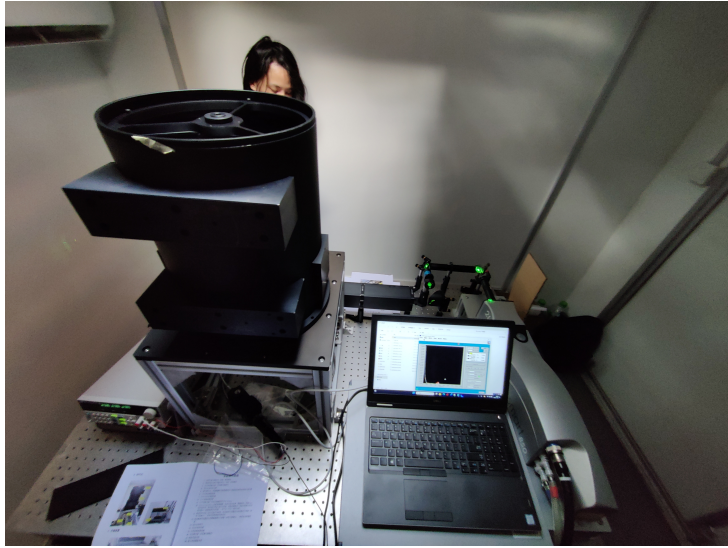


图 2: 收集图

4 数据处理

先使用 LABVIEW 与 Advanced Viewer 将二进制文件转为十进制文件，如图。

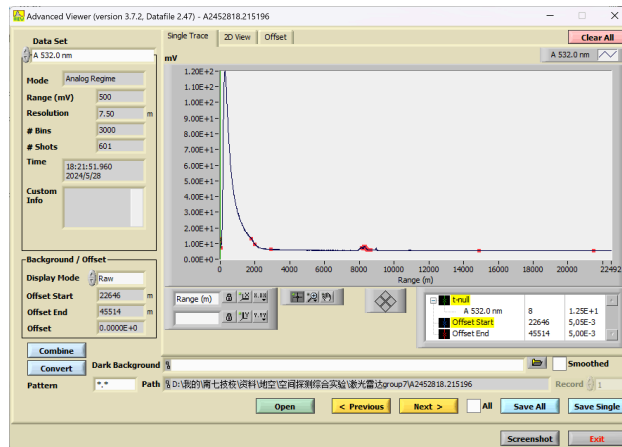


图 3: Advanced Viewer 处理

inclusion.py 将所有十进制文件转为一个集成的 inclusive.csv。

```
# inclusion.py
import os
import pandas as pd

source_folder = 'group7e'
destination_file = 'inclusive.csv'

combined_data = pd.DataFrame()

for file in os.listdir(source_folder):
    if file.endswith('.csv'):
        file_path = os.path.join(source_folder, file)
        data = pd.read_csv(file_path, header=None)

        time_label = os.path.splitext(file)[0]
        data.iloc[:, 1] = range(1, len(data) + 1)

        df = pd.DataFrame({
            'Time': [time_label] * len(data),
            'Height': data.iloc[:, 1],
            'Value': data.iloc[:, 0]
        })
        combined_data = pd.concat([combined_data, df], ignore_index=True)

combined_data.sort_values(by=['Time', 'Height'], inplace=True)

pivoted_data = combined_data.pivot(index='Height', columns='Time', values='Value')
pivoted_data.to_csv(destination_file)
```

kletting.py 反演并输出结果到 results.csv。

```
# kletting.py
import pandas as pd
import numpy as np
import pywt

file_path = 'inclusive.csv'
data = pd.read_csv(file_path)

heights = data.iloc[:, 0]
times = data.columns[1:].astype(int)

data_values = data.iloc[:, 1:].to_numpy()

h = 0.0075
wavelet = 'db4'
level = 5
thr = 0.5
k = 0.86
b = 314
m = 323

def klett_method(column_data, k, b, m, h, wavelet, level, thr):
    coeffs = pywt.wavedec(column_data, wavelet, level=level)
    c_thresh = [pywt.threshold(c, thr, mode='hard') for c in coeffs]
    P = pywt.waverec(c_thresh, wavelet)

    S = np.log(column_data ** 2 * P)

    I = 0
    for t in range(b, m + 1):
```

```

        if t == b or t == m:
            I += h * np.exp((S[t] - S[m]) / k) / 2
        else:
            I += h * np.exp((S[t] - S[m]) / k)

sigma_m = (k * np.exp((S[b] - S[m]) / k) - 1) / (2 * I)

sigma = [0]
for r0 in range(1, 3000):
    J = 0
    for t in range(r0, m + 1):
        if t == b or t == m:
            J += h * np.exp((S[t] - S[m]) / k) / 2
        else:
            J += h * np.exp((S[t] - S[m]) / k)

    sig = np.exp((S[r0] - S[m]) / k) / (1 / sigma_m + 2 * J / k)
    sigma.append(sig)

return sigma

results = []

for i in range(data_values.shape[1]):
    column_data = np.abs(data_values[:, i] - 5.4)
    sigma = klett_method(column_data, k, b, m, h, wavelet, level, thr)
    sigma = abs(np.array(sigma))
    sigma = np.minimum(sigma, 5)
    sigma = np.nan_to_num(sigma, nan=0)
    results.append(sigma)

```

```
results_df = pd.DataFrame(results).transpose()
results_df.columns = times

output_path = 'results.csv'
results_df.to_csv(output_path, index=False)

import matplotlib.pyplot as plt
```

heatmap.py 绘制 heatmap。

```
#heatmap.py
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

file_path = 'results.csv'
data = pd.read_csv(file_path)

time = data.columns.astype(str)
time = [f"18:{t[:2]}:{t[2:4]}" for t in time]
# HEIGHT IS IN KILOMETERS
height = np.arange(data.shape[0]) * 0.0075
sigma_values = data.values[:, :-1]

plt.figure(figsize=(12, 6))
plt.imshow(sigma_values, aspect='auto', cmap='viridis', extent=[0, len(time), height, len(sigma_values)],
plt.colorbar(label='Sigma')
plt.xlabel('Time')
plt.ylabel('Height (km)')
plt.xticks(ticks=np.arange(0, len(time), len(time) // 10), labels=time[:, :len(time) // 10])
plt.title('Extinction Coefficient in km-1')
plt.tight_layout()
plt.show()
```


5 结论

消光系数随高度与时间的分布如图。

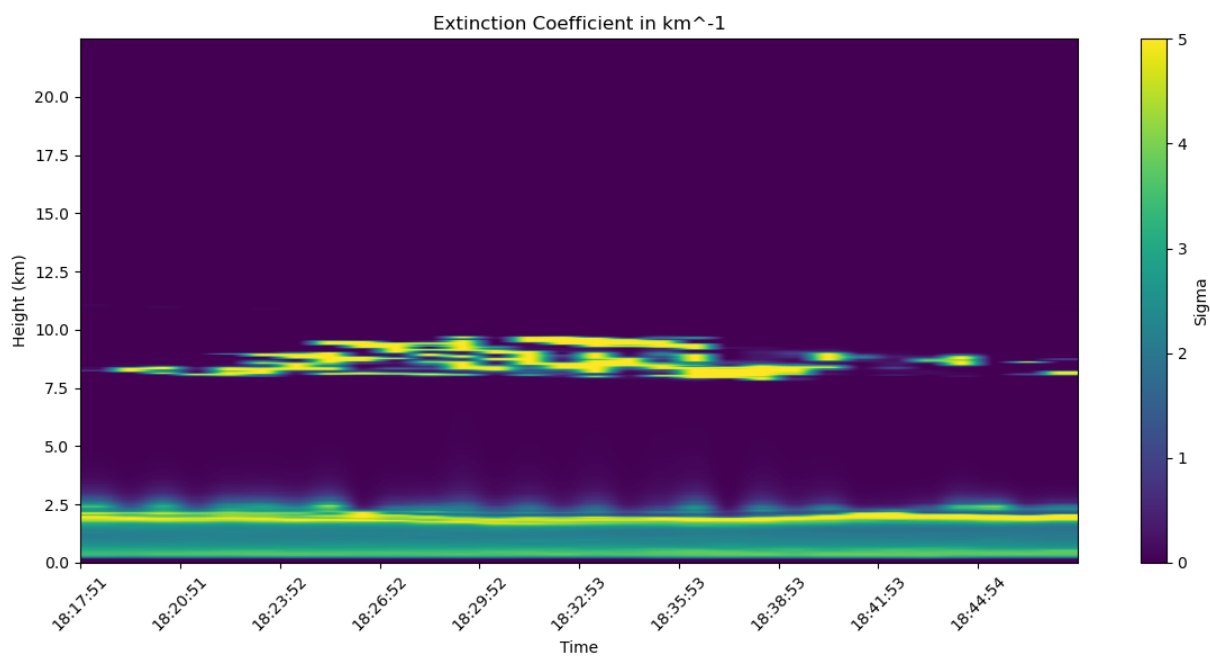


图 4: 消光系数-时间, 高度关系图

在 2.5km 以下, 扬尘与水汽导致地面消光系数较高。约 7.5km 至 10.0km 处, 消光系数有显著的增强, 可由此推断此处含有云层导致的。