之前一直做视觉相关的工作，近期接触激光雷达才发现，激光雷达传感器厂商保存数据的方式一般采用Pcap包的形式，虽然不太清楚为什么要这么做，但不管清不清楚总归只能跟着它这么搞。

一个Pcap文件包括“Pcap报头”，“数据区”两个部分，其中数据区又分成多个数据包，每个包有报头和数据两个部分，总体结构可见下图：

其中固定大小的为“Pcap报头”和数据包中的“数据报头”。

“Pcap报头”大小为24个字节，具体内容因为与本人所要内容无关，直接跳过，等有需要再补充。

“数据报头”大小为16个字节，前8个字节为时间戳数据，9~12字节为数据长度，13~16字节为离线数据长度。如果只想知道数据长度是多少，则只需要读取9~12字节，然后往后面读取其描述的长度即可。

Pcap包的简单解析c++代码如下:

#include <iostream>

#include <opencv2/opencv.hpp>

#include <Windows.h>

#define BUFLEN 1024\*1024

using namespace std;

using namespace cv;

int main()

{

uchar \*data = new uchar[BUFLEN];//开辟数据区

FILE \*pcapfile;

fopen\_s(&pcapfile, "16.pcap","rb");

//读取pcap报头

fread\_s(data, BUFLEN, 1, 24, pcapfile);

//循环读取数据包，直到文件末尾

while (true)

{

//数据包头

int val=fread\_s(data, BUFLEN, 1, 16, pcapfile);

if (!val)

{

break;

}

long packetLen = data[8] | data[9] << 8 | data[10] << 16 | data[11] << 24;//读取数据区长度

//cout << packetLen << endl;

//数据区

val = fread\_s(data, BUFLEN, 1, packetLen, pcapfile);

if (!val)

{

break;

}

//数据处理

}

//对象释放

fclose(pcapfile);

delete[]data;

}

在Windows上，也可以直接用WinPcap来解析，有两种写法：

写法1：

#include <stdio.h>

#include <pcap.h>

#define LINE\_LEN 16

void dispatcher\_handler(u\_char\* temp1,const struct pcap\_pkthdr\* header,const u\_char\* pkt\_data)

{

//打印包信息

printf("%ld:%ld (%ld)\n", header->ts.tv\_sec, header->ts.tv\_usec, header->len);

////打印包内容

//for (int i = 1; (i < header->caplen + 1); i++)

//{

// printf("%.2x ", pkt\_data[i - 1]);

// if ((i % LINE\_LEN) == 0) printf("\n");

//}

//printf("\n\n");

}

int main()

{

pcap\_t\* fp;

char errbuf[PCAP\_ERRBUF\_SIZE];

//打开Pcap包

if ((fp = pcap\_open\_offline("1.pcap",errbuf)) == NULL)

{

return -1;

}

//读取文件直到结尾

pcap\_loop(fp, 0, dispatcher\_handler, NULL);

pcap\_close(fp);

return 0;

}

写法2：

#include <stdio.h>

#include <pcap.h>

#define LINE\_LEN 16

int main()

{

pcap\_t\* fp;

char errbuf[PCAP\_ERRBUF\_SIZE];

//打开Pcap包

if ((fp = pcap\_open\_offline("1.pcap",errbuf)) == NULL)

{

return -1;

}

//读取文件直到结尾

struct pcap\_pkthdr\* header; //报文头

const unsigned char\* pkt\_data; //报文内容

while ((pcap\_next\_ex(fp, &header, &pkt\_data)) > 0)

{

printf("%ld:%ld (%ld)\n", header->ts.tv\_sec, header->ts.tv\_usec, header->len);

}

pcap\_close(fp);

return 0;

}

如果要获得IP信息，可以添加相应的代码如下：

#include <stdio.h>

#include <pcap.h>

#define LINE\_LEN 16

/\* 4 bytes IP address \*/

typedef struct ip\_address

{

u\_char byte1;

u\_char byte2;

u\_char byte3;

u\_char byte4;

}ip\_address;

/\* IPv4 header \*/

typedef struct ip\_header

{

u\_char ver\_ihl; // Version (4 bits) + Internet header length (4 bits)

u\_char tos; // Type of service

u\_short tlen; // Total length

u\_short identification; // Identification

u\_short flags\_fo; // Flags (3 bits) + Fragment offset (13 bits)

u\_char ttl; // Time to live

u\_char proto; // Protocol

u\_short crc; // Header checksum

ip\_address saddr; // Source address

ip\_address daddr; // Destination address

u\_int op\_pad; // Option + Padding

}ip\_header;

int main()

{

pcap\_t\* fp;

char errbuf[PCAP\_ERRBUF\_SIZE];

//打开Pcap包

if ((fp = pcap\_open\_offline("1.pcap",errbuf)) == NULL)

{

return -1;

}

//读取文件直到结尾

struct pcap\_pkthdr\* header; //报文头

const unsigned char\* pkt\_data; //报文内容

ip\_header\* ih;

while ((pcap\_next\_ex(fp, &header, &pkt\_data)) > 0)

{

ih = (ip\_header\*)(pkt\_data + 14); //length of ethernet header

printf("%d.%d.%d.%d -> %d.%d.%d.%d\n",

ih->saddr.byte1,ih->saddr.byte2,ih->saddr.byte3,ih->saddr.byte4,

ih->daddr.byte1,ih->daddr.byte2,ih->daddr.byte3,ih->daddr.byte4);

printf("%ld:%ld (%ld)\n", header->ts.tv\_sec, header->ts.tv\_usec, header->len);

}

pcap\_close(fp);

return 0;

}