AI-VMD/AI People Counting for 360-degree fisheye External Interface Specification

V 1.03

i-PRO Co., Ltd.

Revise Record

VER.	Date	Item no.	Comment	Revise trigger
1.00	2021/11/30	All	First edition	-
1.01	2022/01/27	2.3	-Correct the example of CGI -Changed the name of parameters All.current ->ALL.Current All.list ->ALL.list Area1.current->Area1.Current (same applies to Area2~Area8) -Correct the description of “min”	Document fix
		2.4	-Correct the example of CGI -Changed the example of “mode:latest_comp” -Changed the name of data of csv file count_AB1 -> count_in1 count_BA1 -> count_out1(same applies to Line2~Line8)	
		3.3	Changed the name of data of additional information fromAtoB -> countIn fromBtoA -> countOut	
		5.2	-Changed the description of parameter RightCount: Direction “In” -> Direction “Out” LeftCount: Direction “Out” -> Direction “In” UtnRightCount: Crossed to Direction “In” after crossed to Direction “Out”. ->Crossed to Direction “Out” after crossed to Direction “In”. UtnLeftCount: Crossed to Direction “Out” after crossed to Direction “In”. -> Crossed to Direction “In” after crossed to Direction “Out”. -Correct the value of “AlarmCount” 1~40 -> 1~100	
1.02	2022/04	Cover	Changed the name of a company	Changes in the name of a company
		2.3	Corrected the description of “min”	Document fix
		4.1	Added authentication method	-
1.03	2022/05	Cover	Changed the document title	-
		7.1	Changed the trackID. trackID=3 -> trackID=4	Document fix

Index

1. Abstract

This document describes the interface specification to control the intelligence functions such as AI-VMD/AI People Counting for 360-degree fisheye. It consists of interface commands list, basic sequences.

1.1. Cross Line Counting

This function is that the camera counts the number of people which crossed the lines and stores the counting result to internal memory as CSV files and index file(JPEG). These files can be acquired via http.

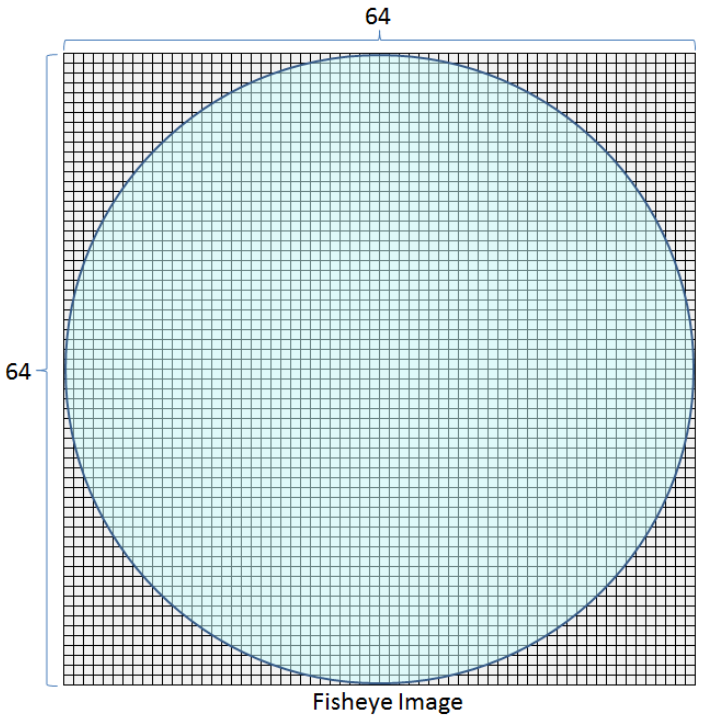
Count information can be obtained via HTTP or as additional information via H.264 / H.265 and JPEG streams. In the case of via http, after changing the time, the data of the time before the change may be sent for a while.

The number of people count will be reset by "storing interval" setting.

1.2. Heat map

This function is that the camera counts the heat map information (counting / loitering) from the image which was divided to 64x64 blocks and the camera stores its result to internal memory as CSV files and index file(JPEG). These files can be acquired http.

The heat map result will be reset by "storing interval" setting.



1.3. Area Counting/Occupancy Detection

This function is that the camera counts the number of people which stayed in the areas and stores the counting result to internal memory as CSV files and index file(JPEG). These files can be acquired via http.

Count information can be obtained via HTTP or as additional information via H.264 / H.265 and JPEG streams. In the case of via http, after changing the time, the data of the time before the change may be sent for a while.

The number of people count will be reset by "storing interval" setting.

1.4. AI-VMD

By using this function, it is possible to detect a moving object and generate an alarm in the following three modes.

INTRUDER: Detects if moving bodies intrude into the predetermined area.
LOITERING: Detects if moving bodies loiter in the predetermined area beyond the fixed time.
CROSS LINE: Detects if moving bodies cross the predetermined line in the direction you specified.

AI-VMD determines whether the detected moving body is a vehicle (four-wheel) or a human.

1.5. Output interface

Output interface of AI-VMD/AI People Counting for 360-degree fisheye is the below.

Function	Output I/F						
	TCP alarm notification	get_io2	Stream Additional information	ONVIF (Analytics Stream)	ONVIF (Event Stream)	CGI (Get the count result)	HTTP POST
Cross Line Counting	—	—	○	○	○	○	○
Heat map	—	—	—	○	○	○	—
Area Counting/ Occupancy detection	○	—	○	○	○	○	○
AI-VMD	○	○	○	○	○	—	—

1.6. Application License

The license of the AI-VMD/AI People Counting for 360-degree fisheye is free.

2. CGI command interface

2.1. Abstract

There are two interface commands to get the count result. The below is the difference between them. They are explained in Chapter 2.3 and 2.4.

How to get count result	Send CGI and get the meta information(JSON file)	Send CGI and download the CSV files
Use case	Check count result in real time	Collect and view count result sometimes (once a week or month etc)
Available function	・Cross Line Counting ・Heat map ・Area Counting/Occupancy detection	・Cross Line Counting ・Area Counting/Occupancy detection
Data collection method	Auto / Manual	Manual
Display in real time	Supported	Not supported

2.2. How to check the application

You can check whether the application is already installed to camera or not by sending CGI as follows.

1. Enter the following URL with camera running.
   http://(Camera IP address) address)/cgi-bin/getinfo?FILE=1

2. You can confirm that the AI-VMD/AI People Counting for 360-degree fisheye is installed from the following message.

EXTAPP1= AI-VMD and AI People Counting EXTAPP2= EXTAPP3=

Depending on the order of installation, "AI-VMD and AI People Counting" may be displayed on EXTAPP2 or EXTAPP3.

2.3. Interface commands (CGI): Get the meta information(JSON files)

[Command interface]
Method: GET

[CGI URL]
http://(Camera IP address) /cgi-bin/adam.cgi?methodName=sendDataToAdamApplication&appName=iVmdApp&s_appDataType=0&s_appData=(base64 data)

[Request parameters]

Parameter	Value	Description
appMethod	get_result	Set the method.
min	1~1440	From the last 24 hours of data stored in the application, information that goes back “min” (in units of 1 minute) from the timing when the CGI is received is returned as the response for the specified number (n : 1~1440). If the camera or application is rebooted, the saved data will be deleted.

The configuration data is JSON format.
When using it, base64 encode the following configuration values. Parameter “min” and its value must be surrounded with "".
{{appMethod:get_result},{"min":"xx"}}

e.g.) min=5
Encode the following configuration data into base64
Original data : {{appMethod:get_result},{"min":"5"}}
Encoded : e3thcHBNZXRob2Q6Z2V0X3Jlc3VsdH0seyJtaW4iOiI1In19
CGI :
http://(Camera IP address)/cgi-bin/adam.cgi?methodName=sendDataToAdamApplication&appName=iVmdApp&s_appDataType=0&s_appData=e3thcHBNZXRob2Q6Z2V0X3Jlc3VsdH0seyJtaW4iOiI1In19

2.3.1. JSON format

Response is returned in the following format.

```
{
  "CameraIPaddress":"xxx.xxx.xxx.xxx",
 "Time":"xxxx/xx/xx xx:xx:xx",
 "TimeZone":"xxxx",
 "SummerTime":x,
 "ALL":[
      {"list": [["xxxx/xx/xx xx:xx:xx", xx, xx]]},
      {"Current":xx}
 ],
 "Area1":[
    {"list": [["xxxx/xx/xx xx:xx:xx", xx, xx]]},
    {"Current":xx}
 ],
  "Area2":[
    {"list": [["xxxx/xx/xx xx:xx:xx", xx, xx]]},
    {"Current":xx}
   ],
   ~
   "Area8":[
    {"list": [["xxxx/xx/xx xx:xx:xx", xx, xx]]},
    {"Current":xx}
   ],
"Line1":[
    {"list": [["xxxx/xx/xx xx:xx:xx", xx, xx]]},
],
  "Line2":[
    {"list": [["xxxx/xx/xx xx:xx:xx", xx, xx]]},
   ],
   ~
   "Line8":[
    {"list": [["xxxx/xx/xx xx:xx:xx", xx, xx]]},
   ]
}
```

【Response Parameter】

Parameter	Value	Notation	Description
CameraIPaddress	(0~255).(0~255). (0~255).(0~255)	Decimal number	Camera IP address (Text type: half-width alphanumeric characters)
Time	Date and time(UTC)		Date and time Format: yyyy/mm/dd hh:mm:ss e.g.) August 29, 2013 12:35:01 Japan time 2013/08/29 03:35:01
TimeZone	-1200~+1300		Time difference from UTC e.g.) Osaka, Sapporo, Tokyo (time difference of 9 hours) X-TZ : +0900
SummerTime	0, 1		Daylight saving time setting 0:non-daylight saving time

			1:daylight saving time
ALL.list	[“Date and time(UTC)”, Average number of detected people(0~100), Number of people detected on time(0~100)]		Number of people in the entire screen (Text type: half-width numeric characters) [Date and timea(UTC)] Fixed time information for each minute. e.g.) 2021/1/11 9:00 2021/1/11 9:00:00~2021/1/11 9:00:59 [Average number of detected people]: Average number of detected users per [Date and time (UTC)]. (Average number of people detected between xx:xx:00 - xx:xx:59) [Number of people detected on time] Number of people detected on time (The number of people detected at the moment of xx:xx:01)
ALL.Current	0~100		Number of people detected in real time for the entire screen (Text type: half-width alphanumeric characters)
Area1.list Area2.list Area3.list Area4.list Area5.list Area6.list Area7.list Area8.list	[“Date and time(UTC)”, Average number of detected people(0~100), Number of people detected on time(0~100)]		Number of people in the entire screen (Text type: half-width numeric characters) [Date and timea(UTC)] Fixed time information for each minute. e.g.) 2021/1/11 9:00 2021/1/11 9:00:00~2021/1/11 9:00:59 [Average number of detected people]: Average number of detected users per [Date and time (UTC)]. (Average number of people detected between xx:xx:00 - xx:xx:59) [Number of people detected on time] Number of people detected on time (The number of people detected at the moment of xx:xx:01)
Area1.Current Area2.Current Area3.Current Area4.Current Area5.Current Area6.Current Area7.Current Area8.Current	0~100		Number of people detected in real time for earch detection area (Text type: half-width alphanumeric characters)
Line1.list Line2.list Line3.list Line4.list Line5.list Line6.list Line7.list Line8.list	[“Date and time(UTC)”, Number of detected people direction “In”(0~100),Number of detected people direction “Out”(0~100)]		Number of people crossing line. (Text type: half-width numeric characters) [Date and time(UTC)] Fixed time information for each minute. e.g.) 2021/1/11 9:00 2021/1/11 9:00:00~2021/1/11 9:00:59 [Number of detected people]: Number of people detected for 1 minute.

* It does not include information on the time when the detection area or line was not set or was invalid.

(Abnormal)
【Response Parameters】

Return value name	Description
faultCode	Error code
faultString	Error string

"400" Bad Request

faultCode="1"	faultString="Invalid Parameter” s_appData cannot be base64 decoded.
faultCode="4"	faultString="Invalid Process ID" The additional application with the specified process identification ID is not running.
faultCode="10"	faultString="Invalid Protocol" There is an error in the argument.
faultCode="25"	faultString="Invalid Application Name" The specified application name is invalid.

"409" Conflict

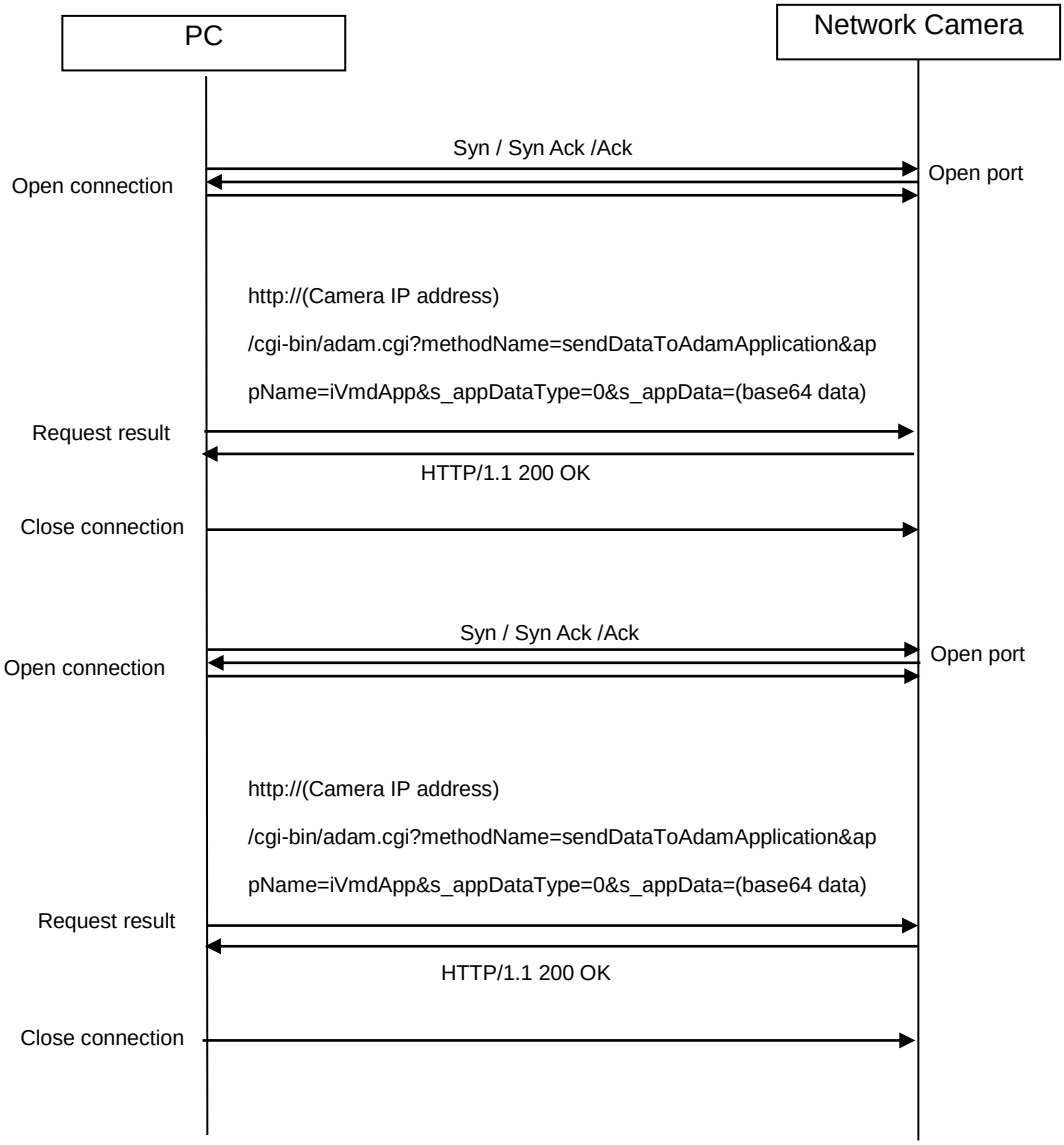
faultCode="13"	faultString="Bad Application Status” Data cannot be accepted because the specified application is in the process of starting or stopping, etc.

"500" Internal Server Error

faultCode="14"	faultString="File Access Error" Internal error (file access error)
faultCode="15"	faultString="I/O error" Internal error (I/O error)
faultCode="16"	faultString="Not Enough Memory" Internal error (insufficient memory)
faultCode="18"	faultString="Internal Error" Internal error (other error)

2.3.2. Getting JSON file

Sequence



Response format

e.g.)

min = {3}, detection area = {area 1}, effective detection area = {area 1}, detection line = {line 1}, effective detection line = {line 1}

Transmission time: {JST 2021/1/11 20:05:40}


```
{
  "CameraIPaddress":"192.168.0.10",
  "Time":"2021/1/11 11:05:40",
  "TimeZone":"+0900",
  "SummerTime":0,
   "ALL":[
      {"list": [["2021/1/11 11:00", 8, 7],
                ["2021/1/11 11:01", 9, 8],
                ["2021/1/11 11:02", 10, 9]
                ]
        },
        {"Current":12}
 ],
  "Area1":[
      {"list": [["2021/1/11 11:00", 5, 4],
                ["2021/1/11 11:01", 7, 6],
                ["2021/1/11 11:02", 8, 6]
                ]
        },
        {"Current":7}
   ],
   "Area2":[
      {"list": []},
      {"Current":0}
   ],
~
   "Area8":[
     {"list":[]},
     {"Current":0}
     ],
"Line1":[
      {"list": [["2021/1/11 11:00", 5, 4],
                ["2021/1/11 11:01", 7, 6],
                ["2021/1/11 11:02", 8, 6],
                ]
        },
   ],
   "Line2":[
      {"list": []},
   ],
~
   "Line8":[
     {"list":[]},
   ]
}
```

2.4. Interface commands (CGI): Download the CSV files

[Command interface]
Method: GET

[CGI URL]
http://(Camera IP address)/cgi-bin/adam.cgi?methodName=sendDataToAdamApplication&appName=iVmdApp&s_appDataType=0&s_appData=(base64 data)

[Request parameters]

Parameter	Value	Description
appMethod	csv	Set the method.
kind	movcnt movcnt_info heatmap_mov heatmap_mov_info heatmap_loi heatmap_loi_info areacnt areacnt_info	Kind of the csv data movcnt: Line cross count result and index images. movcnt_info:Line cross count result heatmap_mov: Heat map information (count) and index images heatmap_mov_info: Heat map information (count) heatmap_loi: Heat map (loitering) information and index images. heatmap_loi_info: Heat map (loitering) information areacnt: Area count result and index images areacnt_info: Area count result *This parameter can't be omitted
mode	range multi latest latest_comp	Kind of response range: Get storing period multi: Get csv/index files by a date. latest: Get latest csv/index file latest_comp: Get latest and completed csv/index file *This parameter can't be omitted *"latest_comp" is valid when "kind" is any one of "heatmap_mov, heatmap_mov_info, heatmap_loi, heatmap_loi_info".
year	(numerical value) (4 columns)	Date of the file to acquire (Year) *This parameter can't be omitted when the 'mode' parameter set to 'multi'.
month	1 - 12	Date of the file to acquire (Month) *This parameter can't be omitted when the 'mode' parameter set to 'multi'.

date	1 - 31	Date of the file to acquire (Date) *This parameter can't be omitted when the 'mode' parameter set to 'multi'.
days	1 - 6	The days of the file to acquire *This parameter can't be omitted when the 'mode' parameter set to 'multi'.
hour	0 - 23	The time of file to acquire. *This parameter can't be omitted when the 'mode' parameter set to 'multi'.

[Command examples]

Getting the storing period of the Line cross count result

http://192.168.0.10/cgi-bin/adam.cgi?methodName=sendDataToAdamApplication&appName=iVmdApp&s_appDataType=0&s_appData={{appMethod:csv},{kind:movcnt},{mode:range}}

*{{appMethod:csv},{kind:movcnt},{mode:range}} is base64 data.

Getting the Line cross count result and index files of 5 days(7/30/2021 00:00 - 8/4/2021 00:00).

(1) In case of Shingapore, the UTC time difference is +8 hours. Specify {date:29},{hour:16} which is 8 hours before 7/30 00:00.

http://192.168.0.10/cgi-bin/adam.cgi?methodName=sendDataToAdamApplication&appName=iVmdApp&s_appDataType=0&s_appData={{appMethod:csv},{kind:movcnt},{mode:multi},{year:2021},{month:7},{date:29},{days:5},{hour:16}}

*{{appMethod:csv},{kind:movcnt},{mode:multi},{year:2021},{month:7},{date:29},{days:5},{hour:16}} is base64 data.

(2) In case of Washington, the UTC time difference is -5 hours. Specify {date:30},{hour:5} which is 5 hours after 7/30 00:00.

http://192.168.0.10/cgi-bin/adam.cgi?methodName=sendDataToAdamApplication&appName=iVmdApp&s_appDataType=0&s_appData={{appMethod:csv},{kind:movcnt},{mode:multi},{year:2021},{month:7},{date:30},{days:5},{hour:5}}

*{{appMethod:csv},{kind:movcnt},{mode:multi},{year:2021},{month:7},{date:30},{days:5},{hour:5}} is base64 data.

Getting the latest Line cross count result and index files.

http://192.168.0.10/cgi-bin/adam.cgi?methodName=sendDataToAdamApplication&appName=iVmdApp&s_appDataType=0&s_appData={{appMethod:csv},{kind:movcnt},{mode:latest}}

*{{appMethod:csv},{kind:movcnt},{mode:latest}} is base64 data.

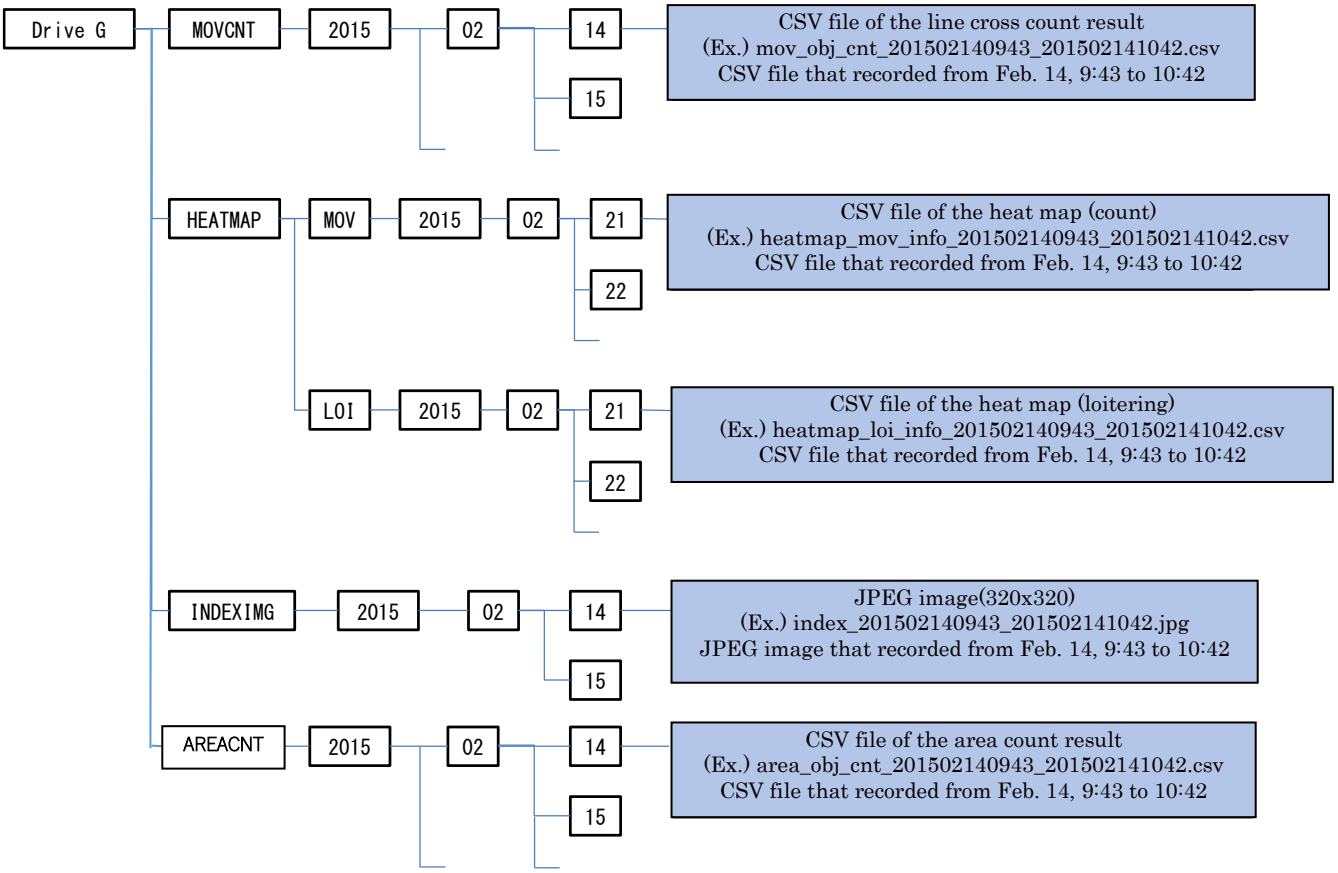
Getting the latest and completed csv/index files.

http://192.168.0.10/cgi-bin/adam.cgi?methodName=sendDataToAdamApplication&appName=iVmdApp& s_appDataType=0&s_appData={{appMethod:csv},{kind:movcnt},{mode:latest_comp}}

*{{appMethod:csv},{kind:movcnt},{mode:latest_comp}} is base64 data.
For the response format and sequence, refer to chapter 2.4.3.

2.4.1. Directory structure and file name

CSV file and index file (JPEG) are stored in the internal memory of the camera. The content of internal memory can be seen as a drive G.



2.4.2. CSV file format

2.4.2.1. Cross Line Counting

The following is a csv file format of the line cross count.

```
s_yyyymmdd,s_hhmm,e_yyyymmdd,e_hhmm,p_hhmm,timezone,summertime
s_x1,s_y1,e_x1,e_y1,count_in1,count_out1
s_x2,s_y2,e_x2,e_y2,count_in2,count_out2
s_x3,s_y3,e_x3,e_y3,count_in3,count_out3
s_x4,s_y4,e_x4,e_y4,count_in4,count_out4
s_x5,s_y5,e_x5,e_y5,count_in5,count_out5
s_x6,s_y6,e_x6,e_y6,count_in6,count_out6
s_x7,s_y7,e_x7,e_y7,count_in7,count_out7
s_x8,s_y8,e_x8,e_y8,count_in8,count_out8
```

Data	Format	Description
s_yyyymmdd	YYYYMMDD YYYY: year(4 columns) MM: month(2 columns) DD: day(2 columns)	Time and date that the camera started to count the number of people from in this csv. (year/month/day) UTC time
s_hhmm	HHmm HH: hour(2columns) mm: minute(2 columns)	Time and date that the camera started to count the number of people from in this csv. (hour/minute) UTC time
e_yyyymmdd	YYYYMMDD YYYY: year(4 columns) MM: month(2 columns) DD: day(2 columns)	Time and date that the csv was closed by. (year/month/day) UTC time
e_hhmm	HHmm HH: hour(2columns) mm: minute(2 columns)	Time and date that the csv was closed by. (hour/minute) UTC time
p_hhmm	HH:mm HH: hour(2columns) mm: minute(2 columns)	Storing interval of the number of people e.g.) In case of 15min-> 00:15
timezone	-12:00 to +12:00 (6 columns)	timezone
summertime	IN, OUT	summertime IN: Daylightsaving time (Summertime) OUT: Not daylight saving time
s_x1	0 to 799	X coordinate of the starting point (Line1)
s_y1	0 to 799	Y coordinate of the starting point (Line1)
e_x1	0 to 799	X coordinate of the ending point (Line1)
e_y1	0 to 799	Y coordinate of the ending point (Line1)
count_in1	0 to 65535	Counting result of the number of people. Direction “In” (Line1)
count_out1	0 to 65535	Counting result of the number of people. Direction “Out” (Line1)

* The above descriptions are common in other lines(s_x2,s_y2,,,, e_x12,e_y12).
* When a line wasn't set, both coordinates information set to (0, 0).

2.4.2.2.Heat map

The following is a csv file format of the heat map.

```
s_yyyymmdd,s_hhmm,e_yyyymmdd,e_hhmm,p_hhmm,timezone,summertime
info(1,1),info(2,1),info(3,1),    ,,,, info(63,1),info(64,1)
info(1,2),info(2,2),info(3,2),    ,,,, info(63,2),info(64,2)
info(1,3),info(2,3),info(3,3),    ,,,, info(63,3),info(64,3)

, , , , , , , ,
, , , , , , , ,
info(m,n),info(m,n),info(m,n),    ,,,, info(m,n),info(m,n)
, , , , , , , ,

info(1,63),info(2,63),info(3,63),    ,,,,,    info(63,63),info(64,63)
info(1,64),info(2,64),info(3,64),    ,,,,,    info(63,64),info(64,64)
```

Data	Format	Description
s_yyyymmdd	YYYYMMDD YYYY: year(4 columns) MM: month(2 columns) DD: day(2 columns)	Time and date that the camera started to count the heat map information from in this csv. (year/month/day) UTC time
s_hhmm	HHmm HH: hour(2columns) mm: minute(2 columns)	Time and date that the camera started to count the heat map information from in this csv. (hour/minute) UTC time
e_yyyymmdd	YYYYMMDD YYYY: year(4 columns) MM: month(2 columns) DD: day(2 columns)	Time and date that the csv was closed by. (year/month/day) UTC time
e_hhmm	HHmm HH: hour(2columns) mm: minute(2 columns)	Time and date that the csv was closed by. (hour/minute) UTC time
p_hhmm	HH:mm HH: hour(2columns) mm: minute(2 columns)	Storing interval of the heat map information e.g.) In case of 15min-> 00:15
timezone	-12:00 to +12:00 (6 columns)	timezone
summertime	IN, OUT	summertime IN: Daylightsaving time (Summertime) OUT: Not daylight saving time
info(m,n)	0 to 65535	Heat map information of (m,n)

2.4.2.3. Area Counting/Occupancy detection

The following is a csv file format of the Area Counting/Occupancy detection.

```
s_yyyymmdd,s_hhmm,e_yyyymmdd,e_hhmm,p_hhmm,timezone,summertime
x1_1,y1_1,x2_1,y2_1, … , x15_1,y15_1,x16_1,y16_1,count_1
x1_2,y1_2,x2_2,y2_2, … , x15_2,y15_2,x16_2,y16_2,count_2
x1_3,y1_3,x2_3,y2_3, … , x15_3,y15_3,x16_3,y16_3,count_3
x1_4,y1_4,x2_4,y2_4, … , x15_4,y15_4,x16_4,y16_4,count_4
x1_5,y1_5,x2_5,y2_5, … , x15_5,y15_5,x16_5,y16_5,count_5
x1_6,y1_6,x2_6,y2_6, … , x15_6,y15_6,x16_6,y16_6,count_6
x1_7,y1_7,x2_7,y2_7, … , x15_7,y15_7,x16_7,y16_7,count_7
x1_8,y1_8,x2_8,y2_8, … , x15_8,y15_8,x16_8,y16_8,count_8
```

Data	Format	Description
s_yyyymmdd	YYYYMMDD YYYY: year(4 columns) MM: month(2 columns) DD: day(2 columns)	Time and date that the camera started to count the number of people from in this csv. (year/month/day) UTC time
s_hhmm	HHmm HH: hour(2columns) mm: minute(2 columns)	Time and date that the camera started to count the number of people from in this csv. (hour/minute) UTC time
e_yyyymmdd	YYYYMMDD YYYY: year(4 columns) MM: month(2 columns) DD: day(2 columns)	Time and date that the csv was closed by. (year/month/day) UTC time
e_hhmm	HHmm HH: hour(2columns) mm: minute(2 columns)	Time and date that the csv was closed by. (hour/minute) UTC time
p_hhmm	HH:mm HH: hour(2columns) mm: minute(2 columns)	Storing interval of the number of people e.g.) In case of 15min-> 00:15
timezone	-12:00 to +12:00 (6 columns)	timezone
summertime	IN, OUT	summertime IN: Daylightsaving time (Summertime) OUT: Not daylight saving time
x1_1	0 to 799	X coordinate of the 1st vertex of angle (Area1)
y1_1	0 to 799	Y coordinate of the 1st vertex of angle (Area1)
count_1	0 to 65535	Counting result of the number of people. (Area1)

* The above descriptions are common in other vertex and areas(x1_2,y1_2,,,, x16_8,y16_8).
* When an area wasn't set, both coordinates information set to (0, 0).

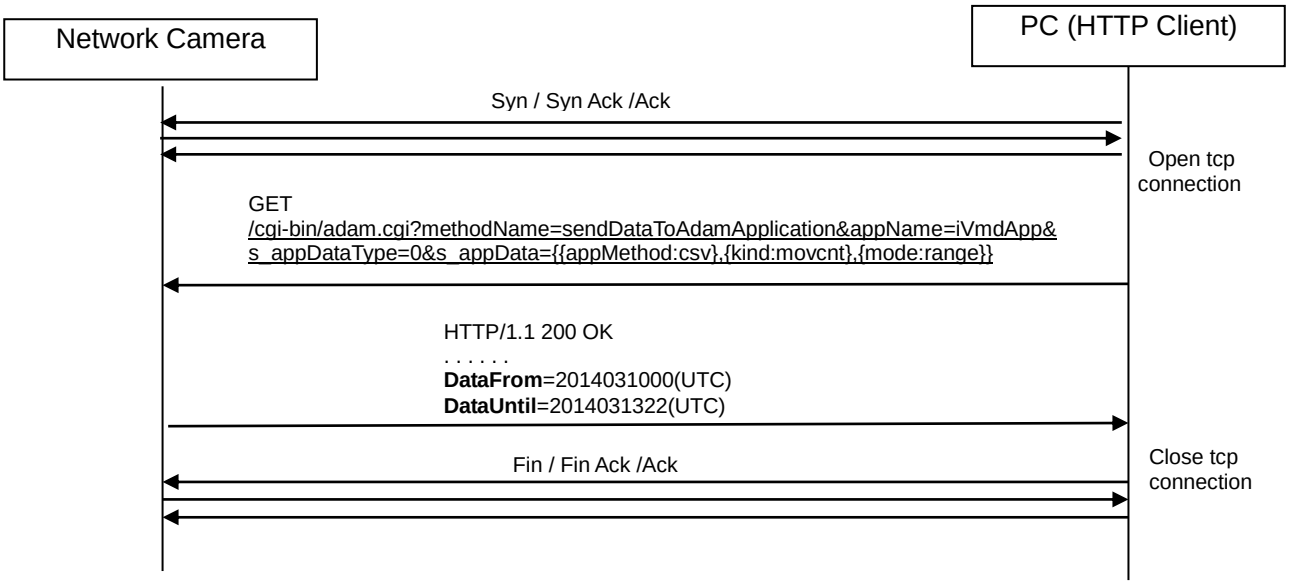
2.4.3. Getting the CSV file

2.4.3.1. About this chapter

The command sequence to acquire the csv file and index file is described in this chapter. In chapter 2.4.3.2 to 2.4.3.4, the command sequence of Cross Line Counting is described. These sequences are also same as Heat map and Area Counting/Occupancy detection functions. In chapter 2.4.3.5, the command sequence of Heat map is described.

2.4.3.2. Get storing period (mode:range)

Sequence



Response format

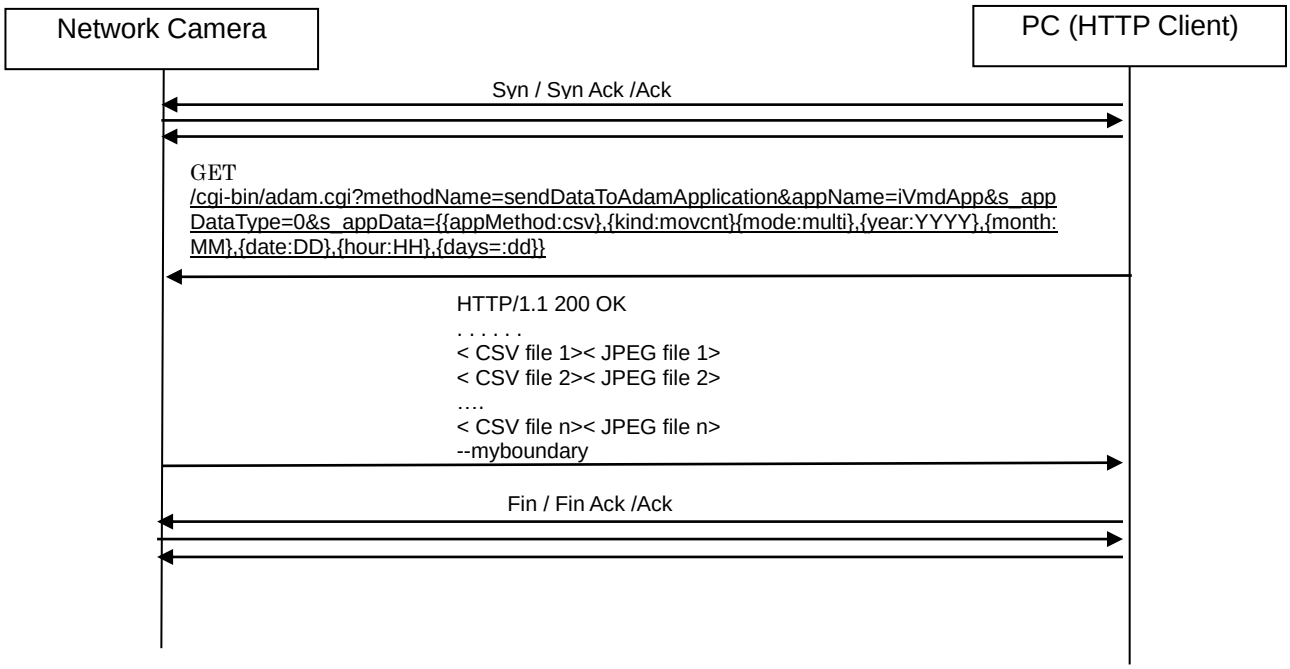
```
HTTP1.1 2000K [CR][LF]
Status 200[CR][LF]
. . . .
Content-Length: xxxxx[CR][LF]
DataFrom=YYYYMMDDHHmm(UTC)[CR][LF]
DataUntil=YYYYMMDDHHmm(UTC)[CR][LF]
[CR][LF]
```

Response data

Data	Format	Description
DataFrom	YYYYMMDDHHmm(UTC) YYYY: year(4 columns) MM: month(2 columns) DD: day(2 columns) HH: hour(2 columns) mm: minute(2 columns)	Time and date of the oldest csv file (UTC time)
DataUntil	YYYYMMDDHHmm(UTC) YYYY: year(4 columns) MM: month(2 columns) DD: day(2 columns) HH: hour(2 columns) mm: minute(2 columns)	Time and date of the latest csv file (UTC time)

2.4.3.3.Get csv/index file by a date (mode:multi)

Sequence



Response format

(In case of kind:movcnt)

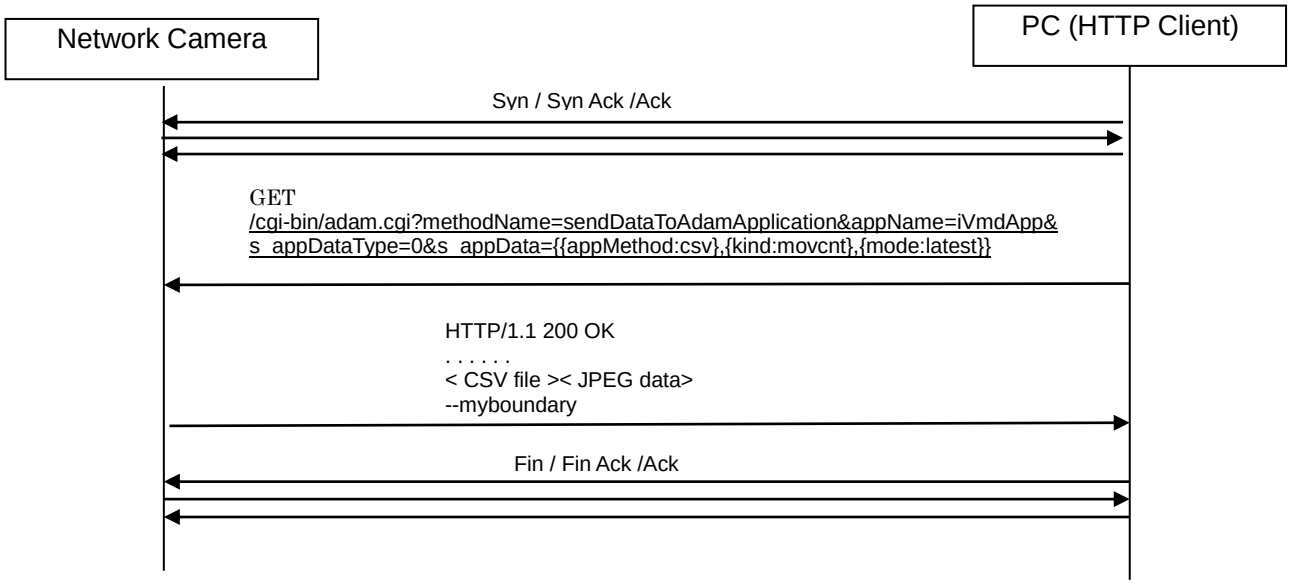
```
HTTP/1.1 200 OK[CR][LF]
Status: 200[CR][LF]
Connection: close[CR][LF]
Content-type: multipart/form-data; boundary=myboundary[CR][LF]
[CR][LF]--myboundary[CR][LF]
Content-Disposition: form-data; name="data" filename="mov_obj_cnt_YYYYMMDDHHMM_yyyymmddhhmm.csv"[CR][LF]
Content-Type: text/plain[CR][LF][CR][LF]
Content-Length: xxxx[CR][LF][CR][LF]
< CSV file 1 >
[CR][LF]--myboundary[CR][LF]
Content-Disposition: form-data; name="data" filename="index_YYYYMMDDHHMM_yyyymmddhhmm.jpg"[CR][LF]
Content-Type: image/jpeg[CR][LF][CR][LF]
Content-Length: xxxx[CR][LF][CR][LF]
< JPEG data 1 >
[CR][LF]--myboundary[CR][LF]
Content-Disposition: form-data; name="data" filename="mov_obj_cnt_YYYYMMDDHHMM_yyyymmddhhmm.csv"[CR][LF]
Content-Type: text/plain[CR][LF][CR][LF]
Content-Length: xxxx[CR][LF][CR][LF]
< CSV file 2 >
[CR][LF]--myboundary[CR][LF]
Content-Disposition: form-data; name="data" filename="index_YYYYMMDDHHMM_yyyymmddhhmm.jpg"[CR][LF]
Content-Type: image/jpeg[CR][LF][CR][LF]
Content-Length: xxxx[CR][LF][CR][LF]
< JPEG data 2 >
[CR][LF]--myboundary[CR][LF]
. . . . . .
[CR][LF]--myboundary[CR][LF]
```

(In case of kind:movcnt_info)

```
HTTP/1.1 200 OK[CR][LF]
Status: 200[CR][LF]
Connection: close[CR][LF]
Content-type: multipart/form-data; boundary=myboundary[CR][LF]
[CR][LF]--myboundary[CR][LF]
Content-Disposition: form-data; name="data" filename="mov_obj_cnt_YYYYMMDDHHMM_yyyymmddhhmm.csv"[CR][LF]
Content-Type: text/plain[CR][LF][CR][LF]
Content-Length: xxxx[CR][LF][CR][LF]
< CSV file 1 >
[CR][LF]--myboundary[CR][LF]
Content-Disposition: form-data; name="data" filename="mov_obj_cnt_YYYYMMDDHHMM_yyyymmddhhmm.csv"[CR][LF]
Content-Type: text/plain[CR][LF][CR][LF]
Content-Length: xxxx[CR][LF][CR][LF]
< CSV file 2 >
[CR][LF]--myboundary[CR][LF]
. . . . . .
[CR][LF]--myboundary[CR][LF]
```

2.4.3.4. Get latest csv/index file (mode:latest)

Sequence



Response format

(In case of kind:movcnt)

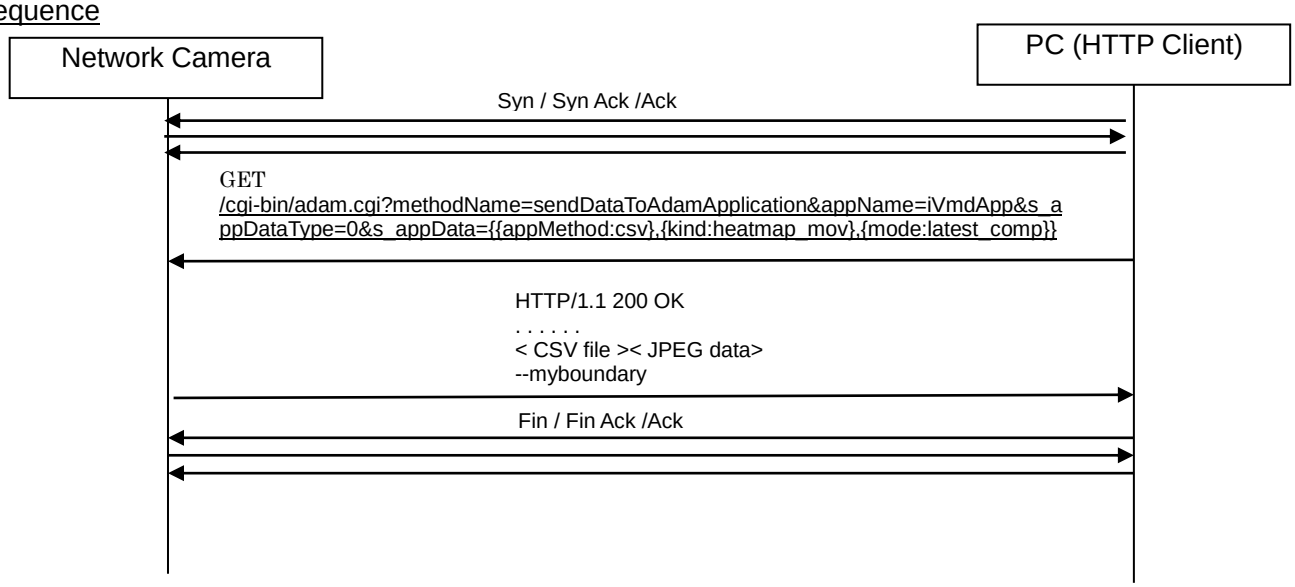
```
HTTP/1.1 200 OK[CR][LF]
Status: 200[CR][LF]
Connection: close[CR][LF]
Content-type: multipart/form-data; boundary=myboundary[CR][LF]
[CR][LF]--myboundary[CR][LF]
Content-Disposition: form-data; name="data" filename="mov_obj_cnt_latest.csv"[CR][LF]
Content-Type: text/plain[CR][LF][CR][LF]
Content-Length: xxxx[CR][LF][CR][LF]
< CSV file >
[CR][LF]--myboundary[CR][LF]
Content-Disposition: form-data; name="data" filename="index_latest.jpg"[CR][LF]
Content-Type: image/jpeg[CR][LF][CR][LF]
Content-Length: xxxx[CR][LF][CR][LF]
< JPEG data >
[CR][LF]--myboundary[CR][LF]
```

(In case of kind:movcnt_info)

```
HTTP/1.1 200 OK[CR][LF]
Status: 200[CR][LF]
Connection: close[CR][LF]
Content-type: multipart/form-data; boundary=myboundary[CR][LF]
[CR][LF]--myboundary[CR][LF]
Content-Disposition: form-data; name="data" filename="mov_obj_cnt_latest.csv"[CR][LF]
Content-Type: text/plain[CR][LF][CR][LF]
Content-Length: xxxx[CR][LF][CR][LF]
< CSV file >
--myboundary[CR][LF]
```

2.4.3.5.Get latest and created csv/index file (mode:latest_comp)

Sequence



Response format

(In case of kind:heatmap_mov)

```
HTTP/1.1 200 OK[CR][LF]
Status: 200[CR][LF]
Connection: close[CR][LF]
Content-type: multipart/form-data; boundary=myboundary[CR][LF]
[CR][LF]--myboundary[CR][LF]
Content-Disposition: form-data; name="data" filename="heatmap_mov_info_latest_comp.csv"[CR][LF]
Content-Type: text/plain[CR][LF][CR][LF]
Content-Length: xxxx[CR][LF][CR][LF]
< CSV file >
[CR][LF]--myboundary[CR][LF]
Content-Disposition: form-data; name="data" filename="index_latest_comp.jpg"[CR][LF]
Content-Type: image/jpeg[CR][LF][CR][LF]
Content-Length: xxxx[CR][LF][CR][LF]
< JPEG data >
[CR][LF]--myboundary[CR][LF]
```

(In case of kind:heatmap_mov_info)

```
HTTP/1.1 200 OK[CR][LF]
Status: 200[CR][LF]
Connection: close[CR][LF]
Content-type: multipart/form-data; boundary=myboundary[CR][LF]
[CR][LF]--myboundary[CR][LF]
Content-Disposition: form-data; name="data" filename="heatmap_mov_info_latest_comp.csv"[CR][LF]
Content-Type: text/plain[CR][LF][CR][LF]
Content-Length: xxxx[CR][LF][CR][LF]
< CSV file >
--myboundary[CR][LF]
```

2.4.3.6. Error response

Error response

```
HTTP1.1 2000K [CR][LF]
Status 200[CR][LF]
. . . .
Content-Length: xxxxx[CR][LF]
xxxxxxxxxx[CR][LF]
[CR][LF]
```

Kind of the error	**Content of xxxxxxxxxx**
A csv file doesn't exist.	No Data.
The function is set "Off" or any line isn't set.	No Data(1).
In preparation (approx. 5 min after starting the function.)	No Data(2).
Other errors	No Data(3).

3. Additional information

3.1. Abstract

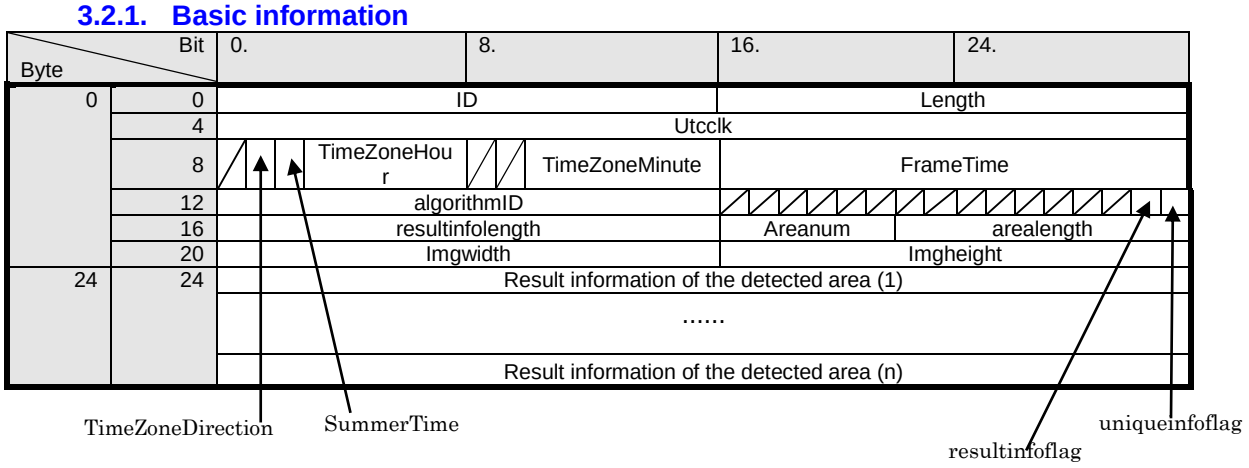
The additional information in the H.264/H.265 rtp header and JPEG header is added when AI-VMD/AI People Counting for 360-degree fisheye is worked. See below for details about location of meta information in streams.

Command_interface_Panasonic_H.265models_verx.xx.pdf

13.7. Locatoin of meta information in each streams

3.2. AI-VMD information

3.2.1. Basic information



Parameter	length(Bit)	Values and comments
ID	16	0x002F (fixed)
Length	16	Total data length of the AI-VMD information (Include 'ID(2byte)' and 'Length(2byte)') (Unit of byte)
Utcclk	32	The career second from 1970 (UTC clock) of the AI-VMD detection information.
TimeZoneDirection	1	The direction of time zone 0 (b) : positive value 1 (b) : negative value
SummerTime	1	0x00 :Not daylight saving time

		0x01 :Daylight saving time (Summer time)
TimeZoneHour	5	Time zone (hour) 0x00: 0hours, 0x01: 1hours, 0x02: 2hours, 0x03: 3hours 0x04: 4hours, 0x05: 5hours, 0x06: 6hours, 0x07: 7hours 0x08: 8hours, 0x09: 9hours, 0x0a: 10hours, 0x0b: 11hours 0x0c: 12hours, 0x0d: 13hours, 0x0e: 14hours, 0x0f: 15hours 0x10: 16hours, 0x11: 17hours, 0x12: 18hours, 0x13: 19hours 0x14: 20hours, 0x15: 21hours, 0x16: 22hours, 0x17: 23hours
TimeZoneMinute	6	Time zone (minute) 0x00: 0minutes, 0x01: 1minutes, 0x02: 2minutes, ... , 0x39: 57minutes, 0x3a: 58minutes, 0x3b: :59minutes
FrameTime	16	Millisecond (Unit of 10 milliseconds) of the AI-VMD detection information. 0x0000: 0 millisecond, 0x0001: 10 milliseconds, ... 0x0062: 980 milliseconds, 0x0063: 990milliseconds
algorithmID	16	0x0000 (fixed)
resultinfoflag	1	Result information flag 0 (b): Not include the result information 1 (b): Include the unique information
uniqueinfoflag	1	0 (b) (fixed)
resultinfolength	16	Length of the Result information (Unit of byte)
Areanum	6	The number of the i-VMD detection Maximum: 0x08
arealength	10	The data length of a result information in each AI-VMD detection. (Unit of byte)
Imgwidth	16	Width of the image for the AI-VMD detection
Imgheight	16	Height of the image for the AI-VMD detection

3.2.2. Result information

Each result information of the detected frame

Byte \ Bit		0.		8.	16.	24.
0	0	areaID			dtctarea	
	4	almtype	dir	almobj		
	8	Hstart			Vstart	
	12	Hcnt			Vcnt	

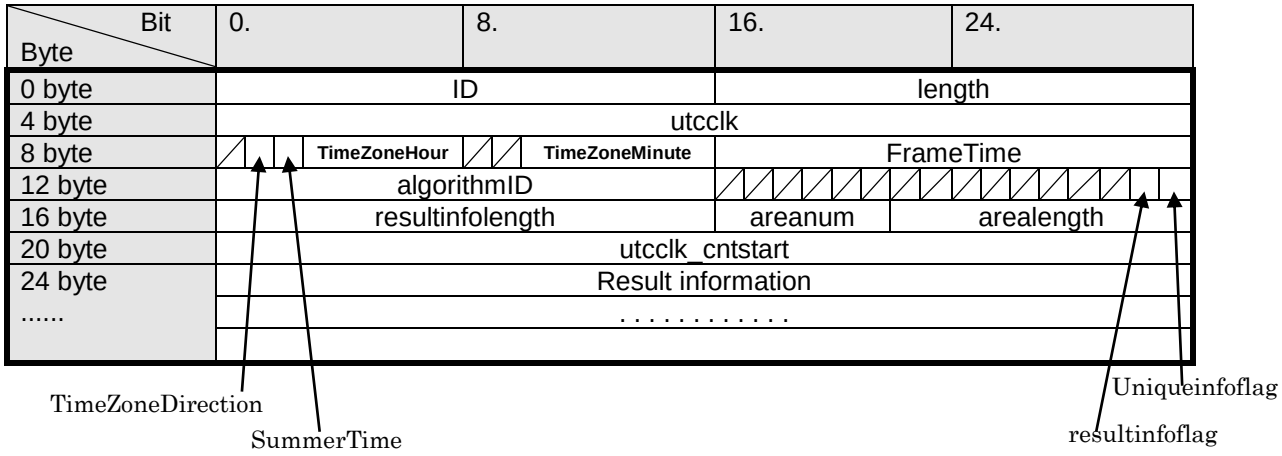
Parameter	length(Bit)	Values and comments
areaID	16	ID of the detected frame 0 to 65535
dtctarea	16	0x0001：Detection program 1 - Detection area 1 0x0002：Detection program 1 - Detection area 2 0x0004：Detection program 1 - Detection area 3 0x0008：Detection program 1 - Detection area 4 0x0010：Detection program 1 - Detection area 5 0x0020：Detection program 1 - Detection area 6 0x0040：Detection program 1 - Detection area 7 0x0080：Detection program 1 - Detection area 8 0x0100：Detection program 2 - Detection area 1 0x0200：Detection program 2 - Detection area 2 0x0400：Detection program 2 - Detection area 3 0x0800：Detection program 2 - Detection area 4

		0x1000：Detection program 2 - Detection area 5 0x2000：Detection program 2 - Detection area 6 0x4000：Detection program 2 - Detection area 7 0x8000：Detection program 2 - Detection area 8
almtype	4	Alarm status 0x01: Intruder detection 0x02: Loitering detection 0x05: Cross line detection 0x0F: Not alarmed
dir	4	Direction for Direction detection/Cross line detection 0x01：Up 0x02：Up-Right 0x03：Right 0x04：Down-Right 0x05：Down 0x06：Down-Left 0x07：Left 0x08：Up-Left 0x09：Direction “In” 0x0a：Direction”Out” 0x0b：Direction “In/Out” 0x00：Not alarmed
almobj	8	Alarmed object 0x01 : Human 0x02 : Vehicle
Hstart	16	X coordinate (Upper left) of the rectangle for the AI-VMD object in 800x800 resolution.
Vstart	16	Y coordinate (Upper left) of the rectangle for the AI-VMD object in 800x800 resolution.
Hcnt	16	Width of the rectangle for the AI-VMD object (Horizontal) in 800x800 resolution
Vcnt	16	Height of the rectangle for the AI-VMD object (Vertical) in 800x800 resolution

AI-VMD information is refreshed every 100 milliseconds.

3.3. Cross Line Counting information

3.3.1. Basic information



Parameter	length(Bit)	Values and comments
ID	16	0x0021 (Fixed)
length	16	Total Data length (include ID and Length) (Unit of byte)
utcclk	32	The career second from 1970 (UTC time)
TimeZoneDirection	1	The direction of time zone 0 (b) : positive value 1 (b) : negative value
SummerTime	1	0 (b) :Not daylight saving time 1 (b) :Daylight saving time (Summer time)
TimeZoneHour	5	Time zone (hour) 0x00: 0hours, 0x01: 1hours, 0x02: 2hours, 0x03: 3hours 0x04: 4hours, 0x05: 5hours, 0x06: 6hours, 0x07: 7hours 0x08: 8hours, 0x09: 9hours, 0x0a: 10hours, 0x0b: 11hours 0x0c: 12hours, 0x0d: 13hours, 0x0e: 14hours, 0x0f: 15hours 0x10: 16hours, 0x11: 17hours, 0x12: 18hours, 0x13: 19hours 0x14: 20hours, 0x15: 21hours, 0x16: 22hours, 0x17: 23hours
TimeZoneMinute	6	Time zone (minute) 0x00: 0minutes, 0x01: 1minutes, 0x02: 2minutes, ... , 0x39: 57minutes, 0x3a: 58minutes, 0x3b: :59minutes
FrameTime	16	Millisecond (Unit of 10 milliseconds) 0x0000: 0 millisecond, 0x0001: 10 milliseconds, ... 0x0062: 980 milliseconds, 0x0063: 990milliseconds
algorithmID	16	Algorithm ID
resultinfoflag	1	Result information flag 0 (b): Not include the result information 1 (b): Include the unique information
uniqueinfoflag	1	0(b) (fixed)
resultinfolength	16	Length of the Result information (Unit of byte)
areanum	6	Number of line.
arealength	10	Data amount by each line (Unit of byte)
utcclk_cntstart	32	Time that started to count the number of people from The career second from 1970(UTC time)
Result information	Variable	People count information

3.3.2. Result information

Bit / Byte	0.	8.	16.	24.
0 byte	areaID			state
4 byte	countIn		countOut	

Parameter	length(Bit)	Values and comments
areaID	16	line ID
state	2	Status 0 :The line is disabled 1 :The line is enabled (direction: In) 2 :The line is enabled (direction: Out) 3 :The line is enabled (both direction: In/Out)
fromAtoB	16	The number of counting people (direction: In)
fromBtoA	16	The number of counting people (direction: Out)

3.4. Area Counting/Occupancy detection information

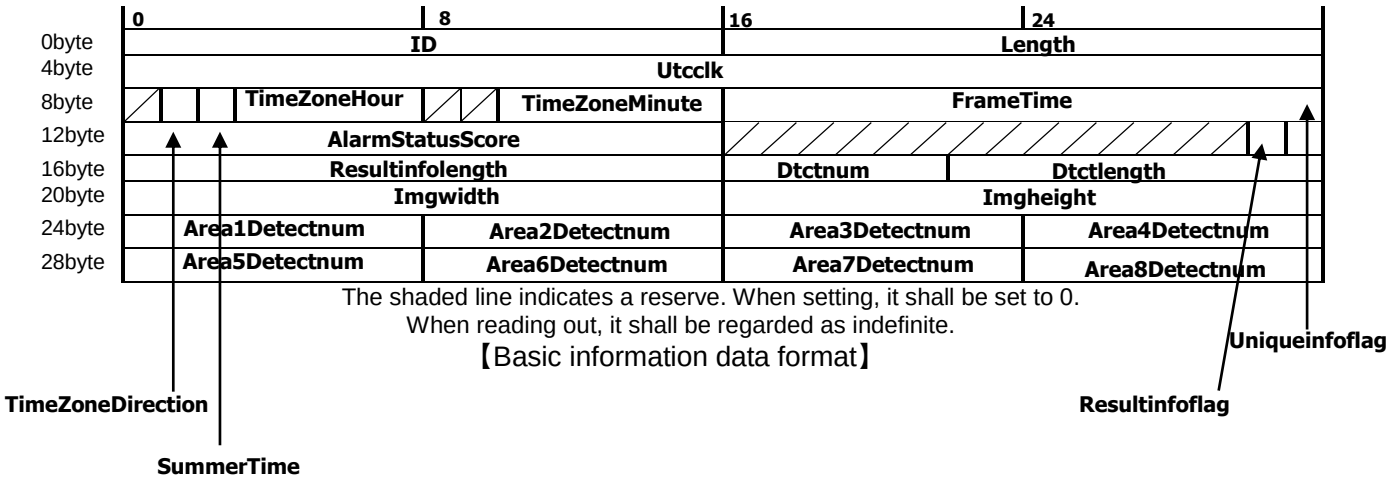
3.4.1. Basic information

The data length of the basic information is fixed at 32 bytes.

【Basic information data】

Parameter	length(Bit)	Values and comments
ID	16	0x0034(ID of additional information)
Length	16	Data length of additional information. Including ID and Length.(byte)
Utcclk	32	UTC clock Total seconds (since 1970)
Reserved	1	Reserved area
TimeZoneDirection	1	0x00: Positive direction 0x01: Negative direction
SummerTime	1	0x00：Winter time 0x01：Summer time
TimeZoneHour	5	0x00：0 hour ～ 0x17：23 hour
Reserved	2	Reserved area
TimeZoneMinute	6	0x00：00 min ～ 0x3B：59 min
FrameTime	16	Counter in units of 10 msec to supplement Utcclk 0x00： 0msec 0x01： 10msec ～ 0x63：990msec
AlarmStatus	16	Alarm information by area 0: Alarm not triggered 1: Alarm triggered

		0bit: Detection area 1 1bit: Detection area 2 2bit: Detection area 3 3bit: Detection area 4 4bit: Detection area 5 5bit: Detection area 6 6bit: Detection area 7 7bit: Detection area 8 e.g.) When an alarm is issued in Area 1 and Area 3 0x0005
Reserved	14	Reserved area
Resultinfoflag	1	Availability of result information (frame information) 0:Not detected 1:Detected
Uniqueinfoflag	1	Availability of unique information(Don't care)
Resultinfolength	16	Data length of the detection frame information.(byte)
Dtctnum	6	Number of detection frames for the entire screen.(Up to 40 frame)
Dtctlength	10	Amount of data per detection frame. (byte)
Imgwidth	16	1920(Image width)
Imgheight	16	1080(Image height)
Area1Detectnum	8	Number of detection frames in detection area 1(Up to 40 frame)
Area2Detectnum	8	Number of detection frames in detection area 2(Up to 40 frame)
Area3Detectnum	8	Number of detection frames in detection area 3(Up to 40 frame)
Area4Detectnum	8	Number of detection frames in detection area 4(Up to 40 frame)
Area5Detectnum	8	Number of detection frames in detection area 5(Up to 40 frame)
Area6Detectnum	8	Number of detection frames in detection area 6(Up to 40 frame)
Area7Detectnum	8	Number of detection frames in detection area 7(Up to 40 frame)
Area8Detectnum	8	Number of detection frames in detection area 8(Up to 40 frame)



The shaded line indicates a reserve. When setting, it shall be set to 0.
When reading out, it shall be regarded as indefinite.

【Basic information data format】

3.4.2. Result information

The result information stores the information of the detected frames.

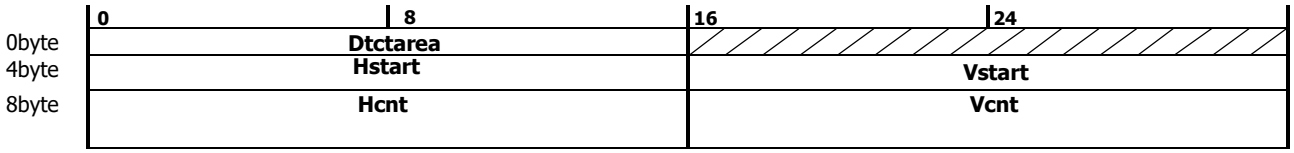
The data length of the result information depends on the number of detected frames and the data length of each frame. Since the number of detected frames and the data length of each frame are Dtctnum and Dtctlength in the basic information, respectively, we can write

Data length of result information = Dtctnum × Dtctlength

in the basic information.

【Result information data】

Parameter	length(Bit)	Values and comments
Dtctarea	16	Belonging detection area 0: Outside 1: Inside 0bit: Detection area 1 1bit: Detection area 2 2bit: Detection area 3 3bit: Detection area 4 4bit: Detection area 5 5bit: Detection area 6 6bit: Detection area 7 7bit: Detection area 8 If the areas overlap, take OR. If the frame does not belong to any area it will become 0x0000. (send frame information that does not belong to any area) e.g.) Area 1 and area 2 0x0003
Reserved	16	Reserved area
Hstart	16	Horizontal starting coordinate of the box (top left) Value converted to FHD coordinate system regardless of input YC resolution
Vstart	16	Vertical starting coordinate of the box (top left) Value converted to FHD coordinate system regardless of input YC resolution
Hcnt	16	Horizontal width of the frame Value converted to FHD coordinate system regardless of input YC resolution
Vcnt	16	Vertical width of the frame Value converted to FHD coordinate system regardless of input YC resolution



The shaded line indicates a reserve. When setting, it shall be set to 0.
When reading out, it shall be regarded as indefinite.

【Result information data format】

4. HTTP periodic transmission

4.1. Telegraphic protocol specification

When using the telegraphic protocol between the AI-VMD/AI People Counting for 360-degree fisheye application(camera) – PC, the HTTP protocol is used. As an HTTP client, the camera sends data to server, such as a processing unit PC.

No.	Item	Specification
1	Number of destination	4
2	Address	Configurable by IPv4 or host name.
3	Port	1～65535
4	Connection	Disconnect the session after each transmission.
5	Content-type	application/json
6	Secure	TLS 1.2
7	Transmission interval	1min,5min,10min,15min,30min,60min e.g.) 5min Transmit every 5 minutes based on 0:00:01 *The transmission time may deviate slightly.
8	Authentication	Digest authentication is used only when user name and password are set.

4.2. Detail of Telegraphic protocol

This section describes the information to be sent from the camera to the server. Common information is stored in the Header section is sent every time the notification interval is set. The number of detected persons to be sent in unit of one minute.

[Common information (Header part)]

Parameter	Value	Description
X-SendTime	Date and Time(UTC)	Date and time Format : [yyyy-mm-dd]T[hh:mm:ss.xx]Z e.g.)JST, August 29, 2013 12:35:01.00 2013-08-29T03:35:01.00Z
X-TZ	-1200~+1300	Time difference from UTC e.g.) Osaka, Sapporo, Tokyo (time difference of 9 hours) X-TZ : +0900
X-ST	0, 1	Daylight saving time setting 0:non-daylight saving time 1:daylight saving time

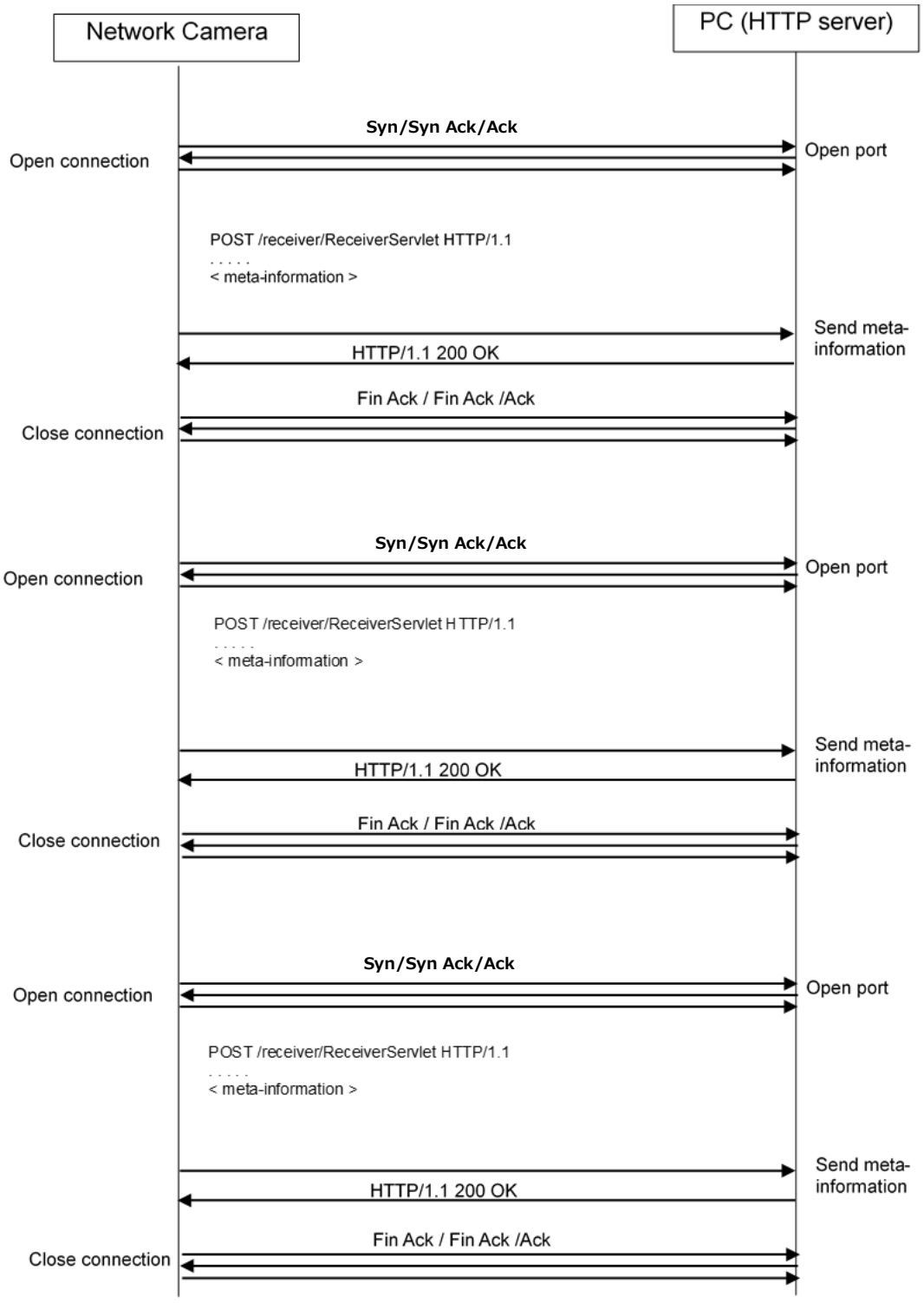
[Number of people information (Body part)]

Parameter	Value	Notation	Description
CameraIPaddress	(0~255).(0~255). (0~255).(0~255)	Decimal number	Camera IP address (Text type: half-width alphanumeric characters)
Time	Date and time(UTC)		Date and time Format: yyyy/mm/dd hh:mm:ss e.g.) August 29, 2013 12:35:01 Japan time 2013/08/29 03:35:01
TimeZone	-1200~+1300		Time difference from UTC e.g.) Osaka, Sapporo, Tokyo (time difference of 9 hours) X-TZ : +0900
SummerTime	0, 1		Daylight saving time setting 0:non-daylight saving time 1:daylight saving time
ALL.list	[“Date and time(UTC)”, Average number of detected people, Number of people detected on time]		Number of people in the entire screen counted by Area Counting. (Text type: half-width numeric characters) [Date and timea(UTC)] Fixed time information for each minute. e.g.) 2021/1/11 9:00 2021/1/11 9:00:00~2021/1/11 9:00:59 [Average number of detected people]: Average number of detected users per [Date and time (UTC)] (Average number of people detected between xx:xx:00 - xx:xx:59) [Number of people detected on time] Number of people detected on time (The number of people detected at the moment of xx:xx:01)
ALL.Current	0~40		Number of people detected in real time for the entire screen. (Text type: half-width alphanumeric characters)
Area1.list Area2.list Area3.list Area4.list Area5.list Area6.list Area7.list Area8.list	[“Date and time(UTC)”, Average number of detected people, Number of people detected on time]		Number of people in the entire screen counted by Area Counting. (Text type: half-width numeric characters) [Date and timea(UTC)] Fixed time information for each minute. e.g.) 2021/1/11 9:00 2021/1/11 9:00:00~2021/1/11 9:00:59 [Average number of detected people]: Average number of detected users per [Date and time (UTC)]. (Average number of people detected between xx:xx:00 - xx:xx:59)

			[Number of people detected on time] Number of people detected on time (The number of people detected at the moment of xx:xx:01)
Area1.Current Area2.Current Area3.Current Area4.Current Area5.Current Area6.Current Area7.Current Area8.Current	0~100		Number of people detected in real time for earch detection area counted by Area Counting. (Text type: half-width alphanumeric characters)
Line1.list Line2.list Line3.list Line4.list Line5.list Line6.list Line7.list Line8.list	[“Date and time(UTC)”, Number of detected people(Direction “In”), Number of detected people(Direction “Out”)]		Number of people detected for earch detection line counted by Cross Line Counting.

*It does not include information about the time when the detection area and line were not set or were disabled.

4.3. Telegraphic protocol sequence



4.4. Transmission format

An example of the transmission format is shown below.

POST /receiver/ReceiverServlet HTTP/1.1[CR][LF] Header part
Content-Length: xxxxx[CR][LF]
User-Agent: i-PRO Camera/1.0[CR][LF]
Connection: close[CR][LF]
Content-type: application/json; charset=utf-8[CR][LF]
X-SendTime: 2021-1-11T11:05:00.00Z[CR][LF]
X-TZ: +0900[CR][LF]
X-ST:0[CR][LF]
[CR][LF]
<meta-information(JSON format)>
Body part

The format of meta-information (Body part) is shown below.

Detection area = {Area 1, 2}, Effective detection area = {Area 1}, Detection line = {Line 1, 2}, Effective detection line = {Line1}, Transmission interval setting = {5min}, Transmission time = {JST 2021/1/11 20:05:00}


```
{
  "CameraIPaddress":"192.168.0.10",
  "CameraIPaddress":"192.168.0.10",
  "Time":"2021/1/11 11:05:00",
  "TimeZone":"+0900",
  "SummerTime":0,
   "ALL":[
      {"list": [["2021/1/11 11:00", 8, 7],
                ["2021/1/11 11:01", 9, 8],
                ["2021/1/11 11:02", 10, 9],
                ["2021/1/11 11:03", 12, 10],
                ["2021/1/11 11:04", 12, 10]
                ]
        },
        {"Current":12}
  ],
  "Area1":[
      {"list": [["2021/1/11 11:00", 5, 4],
                ["2021/1/11 11:01", 7, 6],
                ["2021/1/11 11:02", 8, 6],
                ["2021/1/11 11:03", 9, 8],
                ["2021/1/11 11:04", 6, 6]
                ]
        },
        {"Current":7}
   ],
   "Area2":[
      {"list": []},
      {"Current":0}
   ],
   ～
"Area8":[
     {"list":[]},
   {"Current":0}
   ],
"Line1":[
      {"list": [["2021/1/11 11:00", 5, 4],
                ["2021/1/11 11:01", 7, 6],
                ["2021/1/11 11:02", 8, 6],
                ["2021/1/11 11:03", 9, 8],
                ["2021/1/11 11:04", 6, 6]
                ]
        },
   ],
   "Line2":[
      {"list": []}
   ],
～
  "Line8":[
     {"list":[]},
   ],
}
```

5. ONVIF Meta Stream

There are two kinds of metadata information.

1. Analytics stream : It is sent periodically per 0.10s(5MP camera) or 0.13s(9MP camera).
2. Event stream : It is sent when event occurs. Send interval is different for each function.

5.1. Analytics stream

The format of Analytics stream is common with all finctions of AI-VMD/AI People Counting for 360-degree fisheye.

5.1.1. Parameter description

Parameter	Value	Description
UTC time	Date and Time(UTC)	UTC of video frame
ObjectId	4 byte integer	Detected object ID
BoundingBox	-1 ~ 1	Rectangle area of the detected object (coordinate of upper-left position and lower-right position)
CenterOfGravity	-1 ~ 1	Midpoint of BoundingBox
Class	- Human, Vehicle(object) - 0, 1(Likelihood)	Kind of object and likelihood
DirectionNamed	Up/Right/Down/Left/UpRight/UpLeft/DownRight/DownLeft	Direction of movement
ObjectCount	0 ~ 65,635 (Decimal number)	Sum of the number of object

5.1.2. Meta Stream Example

```
<?xml version="1.0" encoding="utf-8"?>
<tt:MetadataStream xmlns:tt="http://www.onvif.org/ver10/schema">
 <tt:VideoAnalytics>
  <tt:Frame UtcTime="2020-01-20T10:00:08.203Z">
   <tt:Object ObjectId=“101”>
    <tt:Appearance>
     <tt:Shape>
      <tt:BoundingBox left="-0.20" top="0.99" right="0.83" bottom="-0.78" />
      <tt:CenterOfGravity x="0.73" y="0.105" />
     </tt:Shape>
     <tt:Class>
      <tt:ClassCandidate>
       <tt:Type>Human</tt:Type>
       <tt:Likelihood>0.8</tt:Likelihood>
      </ tt:ClassCandidate>
      </tt:Class>
    </tt:Appearance>
        <tt:Extension>
          <Properties>
            <Property name=“DirectionNamed”>Right</Property>
          </Properties>
        </tt:Extension>
   </tt:Object>
     <tt:Extension>
          <Properties>
          <Property name=“ObjectCount”>100</Property>
        </Properties>
     </tt:Extension>
  </tt:Frame>
 <tt:Frame UtcTime="2020-01-20T10:00:18.203Z">
  ... (another frame information)
 </tt:Frame>
</tt:VideoAnalytics>
</tt:MetadataStream>
```

5.2. Event steram

The format of Event stream is different for each function of AI-VMD/AI People Counting for 360-degree fisheye.

5.2.1. AI-VMD

5.2.1.1. Parameter description

Event	Parameter	Value	Description
Intruder	UTC time	Date and Time(UTC)	UTC of video frame
	VideoSource	VideoSourceConfig	Token name of VideoSourceConfig
	Rule	Rule1, Rule2, Rule3, Rule4, Rule5, Rule6, Rule7, Rule8	Area number(Area1 ~ Area8)
	ObjectId	4 byte integer	Detected object ID
	IsInside	true, false	true: Object is in the detection area. false: Object is outside the detection area.
	ClassTypes	Human, Vehicle	Detectoin object
	Image	Base64 encoded	JPEG image when the event occurs
Loitering	Utc time	Date and Time(UTC)	UTC of video frame
	VideoSource	VideoSourceConfig	Token name of VideoSourceConfig
	Rule	Rule1, Rule2, Rule3, Rule4, Rule5, Rule6, Rule7, Rule8	Area number(Area1 ~ Area8)
	ObjectId	4 byte integer	Detected object ID
	Since	Date and Time(UTC)	Starting time of loitering detection
	ClassTypes	Human, Vehicle	Detectoin object
	Image	Base64 encoded	JPEG image when the event occurs
Cross line	Utc time	Date and Time(UTC)	UTC of video frame
	VideoSource	VideoSourceConfig	Token name of VideoSourceConfig

	Rule	Rule1, Rule2, Rule3, Rule4, Rule5, Rule6, Rule7, Rule8	Line number(Line1 ~ Line8)
	ObjectId	4 byte integer	Detected object ID
	ClassTypes	Human, Vehicle	Detectoin object
	Image	Base64 encoded	JPEG image when the event occurs

5.2.1.2.Meta Stream Example

[Event stream of Intruder detection]

```
<?xml version="1.0" encoding="UTF-8"?>
<tt:MetaDataStream xmlns:tt="http://www.onvif.org/ver10/schema"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2">
 <tt:Event>
  <wsnt:NotificationMessage>
   <wsnt:Topic Dialect=http://www.onvif.org/ver10/tev/topicExpression/ConcreteSet
    xmlns:tns1="http://www.onvif.org/ver10/topics"
    xmlns:tnspana1="http://panasonic.co.jp/sn/psn/2010/event/topics">tns1:RuleEngine/Field
    Detector/ObjectsInside
   </wsnt:Topic>
    <wsnt:Message>
     <tt:Message UtcTime="2021-11-15T12:14:26Z">
      <tt:Source>
       <tt:SimpleItem Name="VideoSource" Value="VideoSourceConfig"/>
       <tt:SimpleItem Name="Rule" Value="Rule1"/>
      </tt:Source>
       <tt:Data>
        <tt:SimpleItem Name="IsInside" Value="true"/>
        <tt:SimpleItem Name="ObjectId" Value="2681"/>
        <tt:SimpleItem Name="ClassTypes" Value="Human"/>
         <tt:ElementItem Name="Image">
          <xsd:base64Binary>/9j//gBMAB ((*snip*)) v1/CgR//2Q==</xsd:base64Binary>
         </tt:ElementItem>
       </tt:Data>
      </tt:Message>
     </wsnt:Message>
   </wsnt:NotificationMessage>
 </tt:Event>
</tt:MetaDataStream>
```

[Event stream of loitering detection]

```
<?xml version="1.0" encoding="UTF-8"?>
<tt:MetaDataStream xmlns:tt="http://www.onvif.org/ver10/schema"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2">
 <tt:Event>
  <wsnt:NotificationMessage>
   <wsnt:Topic Dialect=http://www.onvif.org/ver10/tev/topicExpression/ConcreteSet
    xmlns:tns1=http://www.onvif.org/ver10/topics
    xmlns:tnspana1="http://panasonic.co.jp/sn/psn/2010/event/topics">tns1:RuleEngine/Loiter
    ingDetector/ObjectIsLoitering
   </wsnt:Topic>
  <wsnt:Message>
   <tt:Message UtcTime="2021-11-16T03:58:31Z">
    <tt:Source>
     <tt:SimpleItem Name="VideoSource" Value="VideoSourceConfig"/>
     <tt:SimpleItem Name="Rule" Value="Rule1"/>
    </tt:Source>
    <tt:Data>
     <tt:SimpleItem Name="ObjectId" Value="71"/>
     <tt:SimpleItem Name="Since" Value="2021-11-16T03:58:21Z"/>
     <tt:SimpleItem Name="ClassTypes" Value="Human"/>
      <tt:ElementItem Name="Image">
       <xsd:base64Binary>/9j//gBMAB ((*snip*)) v1/CgR//2Q==</xsd:base64Binary>
      </tt:ElementItem>
     </tt:Data>
    </tt:Message>
   </wsnt:Message>
  </wsnt:NotificationMessage>
 </tt:Event>
</tt:MetaDataStream>
```

[Event stream of cross line detection]

```
<?xml version="1.0" encoding="UTF-8"?>
 <tt:MetaDataStream xmlns:tt=http://www.onvif.org/ver10/schema
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2">
  <tt:Event>
   <wsnt:NotificationMessage>
    <wsnt:Topic Dialect=http://www.onvif.org/ver10/tev/topicExpression/ConcreteSet
    xmlns:tns1="http://www.onvif.org/ver10/topics"
    xmlns:tnspana1="http://panasonic.co.jp/sn/psn/2010/event/topics">tns1:RuleEngine/LineD
    etector/Crossed</wsnt:Topic>
   <wsnt:Message>
    <tt:Message UtcTime="2021-11-16T06:42:40Z">
     <tt:Source>
      <tt:SimpleItem Name="VideoSource" Value="VideoSourceConfig"/>
      <tt:SimpleItem Name="Rule" Value="Rule1"/>
     </tt:Source>
     <tt:Data>
      <tt:SimpleItem Name="ObjectId" Value="1064"/>
      <tt:SimpleItem Name="ClassTypes" Value="Human"/>
       <tt:ElementItem Name="Image">
       <xsd:base64Binary>/9j//gBMAB ((*snip*)) v1/CgR//2Q==</xsd:base64Binary>
      </tt:ElementItem>
     </tt:Data>
    </tt:Message>
   </wsnt:Message>
  </wsnt:NotificationMessage>
 </tt:Event>
</tt:MetaDataStream>
```

5.2.2. Cross Line Counting

Send interval is 15s or 1min which is changeable according to setup.

5.2.2.1. Parameter description

Parameter	Value	Description
UTC time	Date and Time(UTC)	UTC of video frame
VideoSource	VideoSourceConfig	The token of VideoSourceConfiguration
Rule	LineCount_Rule1, LineCount_Rule2, LineCount_Rule3, LineCount_Rule4, LineCount_Rule5, LineCount_Rule6, LineCount_Rule7, LineCount_Rule8	Line number
StartTime	Date and Time(UTC)	Starting time of counting(date, hour, minute, second, millisecound)
Count	0~65,535 (Decimal number)	Sum of the number of people crossed the line to Directoin “In” and Direction “Out”.
RightCount	0~65,535 (Decimal number)	Number of people crossed the line to Direction “Out”.
LeftCount	0~65,535 (Decimal number)	Number of people crossed the line to Direction “In”.
UtnRightCount UtnLeftCount	0~65,535 (Decimal number)	Number of people U-turned the line. UtnRightCount : Crossed to Direction “Out” after crossed to Direction “In”. UtnLeftCount : Crossed to Direction “In” after crossed to Direction “Out”.

5.2.2.2. Meta Stream Example

```
<?xml version="1.0" encoding="UTF-8"?>
<tt:MetaDataStream xmlns:tt="http://www.onvif.org/ver10/schema"
xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2">
<tt:Event>
 <wsnt:NotificationMessage>
        <wsnt:Topic Dialect="http://www.onvif.org/ver10/tev/topicExpression/ConcreteSet"
                xmlns:tns1="http://www.onvif.org/ver10/topics">
          tns1:RuleEngine/CountAggregation/Counter
        </wsnt:Topic>
        <wsnt:Message>
          <tt:Message UtcTime="2021-07-02T17:00:00.346Z">
            <tt:Source>
              <tt:SimpleItem Name="VideoSoruce" Value="VideoSourceConfig"/>
              <tt:SimpleItem Name="Rule" Value="LineCount_Rule1"/>
            </tt:Source>
            <tt:Data>
              <tt:SimpleItem Name="StartTime" Value="2021-07-02T00:00:00.000Z"/>
              <tt:SimpleItem Name="RightCount" Value="400"/>
              <tt:SimpleItem Name="LeftCount" Value="600"/>
              <tt:SimpleItem Name="Count" Value="1000"/>
              <tt:SimpleItem Name="UtnRightCount" Value="1"/>
              <tt:SimpleItem Name="UtnLeftCount" Value="2"/>
           </tt:Data>
           </tt:Message>
           <tt:Message UtcTime="2021-07-02T17:00:00.346Z">
            <tt:Source>
              <tt:SimpleItem Name="VideoSoruce" Value="VideoSourceConfig"/>
              <tt:SimpleItem Name="Rule" Value="LineCount_Rule2"/>
            </tt:Source>
           <tt:Data>
            <tt:SimpleItem Name="StartTime" Value="2021-07-02T00:00:00.000Z"/>
            <tt:SimpleItem Name="RightCount" Value="300"/>
            <tt:SimpleItem Name="LeftCount" Value="700"/>
            <tt:SimpleItem Name="Count" Value="1000"/>
            <tt:SimpleItem Name="UtnRightCount" Value="1"/>
            <tt:SimpleItem Name="UtnLeftCount" Value="2"/>
           </tt:Data>
          </tt:Message>
       …
          <tt:Message UtcTime="2021-07-02T17:00:00.346Z">
            <tt:Source>
              <tt:SimpleItem Name="VideoSoruce" Value="VideoSourceConfig"/>
              <tt:SimpleItem Name="Rule" Value="LineCount_Rule8"/>
            </tt:Source>
          <tt:Data>
            <tt:SimpleItem Name="StartTime" Value="2021-07-02T00:00:00.000Z"/>
            <tt:SimpleItem Name="RightCount" Value="400"/>
            <tt:SimpleItem Name="LeftCount" Value="500"/>
            <tt:SimpleItem Name="Count" Value="900"/>
            <tt:SimpleItem Name="UtnRightCount" Value="1"/>
            <tt:SimpleItem Name="UtnLeftCount" Value="2"/>
           </tt:Data>
          </tt:Message>
        </wsnt:Message>
     </wsnt:NotificationMessage>
   </tt:Event>
</tt:MetaDataStream>
```

5.2.3. Heat Map

Send interval is 1min or 5min, 15min which is changeable according to setup.

5.2.3.1. Parameter description

Parameter	Value	Description
UTC time	Date and Time(UTC)	UTC of video frame
VideoSource	VideoSourceConfig	The token of VideoSourceConfiguration
StartTime	Date and Time(UTC)	Starting time of counting(date, hour, minute, second, millisecound)
EndTime	Date and Time(UTC)	Ending time of counting(date, hour, minute, second, millisecound)
Duration	Duration format	Duration of counting
HeatMapMov	Binary data whitch is Base64 encoded	Counting map information. (Number of people in area devided into 64×64 cell)
HeatMapLoi	Binary data whitch is Base64 encoded	Loitering map information (Number of people in area devided into 64×64 cell)
Image	Base64 encoded	Index image which is removed dynamic body

5.2.3.2.Meta Stream Example

```
<?xml version="1.0" encoding="UTF-8"?>
<tt:MetaDataStream xmlns:tt="http://www.onvif.org/ver10/schema"
xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<tt:Event>
 <wsnt:NotificationMessage>
        <wsnt:Topic Dialect="http://www.onvif.org/ver10/tev/topicExpression/ConcreteSet"
              xmlns:tnspana1=”http://panasonic.co.jp/sn/psn/2010/event/topics>
          tns1:RuleEngine/tnspana1:HeatMap
        </wsnt:Topic>
        <wsnt:Message>
          <tt:Message UtcTime="2021-07-02T17:00:00.346Z">
            <tt:Source>
              <tt:SimpleItem Name="VideoSoruce" Value="VideoSourceConfig"/>
            </tt:Source>
            <tt:Data>
             <tt:SimpleItem Name="StartTime" Value="2021-07-15T00:00:00.000Z"/>
             <tt:SimpleItem Name="EndTime" Value="2021-07-15T01:00:00.000Z"/>
             <tt:SimpleItem Name="Duration" Value="PT1H"/>
            <tt:ElementItem Name="HeatMapMov">
             <xsd:base64Binary>NjU1MzUsNj ((*snip*)) U1MzUNCg==</xsd:base64Binary>
            </tt:ElementItem>
            <tt:ElementItem Name="HeatMapLoi">
             <xsd:base64Binary>NjU1MzUsNj ((*snip*)) U1MzUNCg==</xsd:base64Binary>
            </tt:ElementItem>
            <tt:ElementItem Name="Image">
             <xsd:base64Binary>/9j//gBMAB ((*snip*)) v1/CgR//2Q==</xsd:base64Binary>
            </tt:ElementItem>
            </tt:Data>
          </tt:Message>
        </wsnt:Message>
   </wsnt:NotificationMessage>
</tt:Event>
</tt:MetaDataStream>
```

5.2.4. Area Counting/AI Occupancy Detection

Send timing is alarm activation and per 15s or 1 min which is changable according to setup.

5.2.4.1. Parameter description

Parameter	Value	Description
UTC time	Date and Time(UTC)	UTC of video frame
VideoSource	VideoSourceConfig	The token of VideoSourceConfiguration
Rule	AreaCount_Rule1, AreaCount_Rule2, AreaCount_Rule3, AreaCount_Rule4, AreaCount_Rule5, AreaCount_Rule6, AreaCount_Rule7, AreaCount_Rule8	Area number
AlarmStatus	true, false	Type of sending true : Alarm activation false : Sent periodically per 15s or 1 min *AlarmStatus of sending periodically per 15s or 1min will be “true” in case of the area exceed the alarm threshold. *When the alarm notification operation setting is OFF, AlarmStatus is always “false”.
Count	0~65,535 (Decimal number)	Average number of people staying in the area from the last reset to the next transmission
AlarmCount	1~100 (Decimal number)	Alarm condition of headcount
AlarmTime	1~600 [sec] (Decimal number)	Alarm condition of staying time

5.2.4.2. Meta Stream Example

In case of sending periodically per 15s or 1 minute.

```
<?xml version="1.0" encoding="UTF-8"?>
<tt:MetaDataStream xmlns:tt="http://www.onvif.org/ver10/schema"
xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2">
<tt:Event>
  <wsnt:NotificationMessage>
    <wsnt:Topic Dialect="http://www.onvif.org/ver10/tev/topicExpression/ConcreteSet"
                xmlns:tns1=http://www.onvif.org/ver10/topics
                xmlns:tnspana1=”http://panasonic.co.jp/sn/psn/2010/event/topics”>
      tns1:RuleEngine/CountAggregation/tnspana1:AreaCounter
    </wsnt:Topic>
    <wsnt:Message>
      <tt:Message UtcTime="2021-07-02T17:00:00.346Z">
        <tt:Source>
          <tt:SimpleItem Name="VideoSoruce" Value="VideoSourceConfig"/>
          <tt:SimpleItem Name="Rule" Value="AreaCount_Rule1"/>
        </tt:Source>
        <tt:Data>
         <tt:SimpleItem Name="AlarmStatus" Value="false"/>
         <tt:SimpleItem Name="Count" Value="3"/>
         <tt:SimpleItem Name="AlarmCount" Value="5"/>
         <tt:SimpleItem Name="AlarmTime" Value="PT10S"/>
        </tt:Data>
        </tt:Message>
         <tt:Message UtcTime="2021-07-02T17:00:00.346Z">
          <tt:Source>
           <tt:SimpleItem Name="VideoSoruce" Value="VideoSourceConfig"/>
           <tt:SimpleItem Name="Rule" Value="AreaCount_Rule2"/>
          </tt:Source>
         <tt:Data>
          <tt:SimpleItem Name="AlarmStatus" Value="true"/>
          <tt:SimpleItem Name="Count" Value="10"/>
          <tt:SimpleItem Name="AlarmCount" Value="5"/>
          <tt:SimpleItem Name="AlarmTime" Value="PT10S"/>
         </tt:Data>
        </tt:Message>
          …
        <tt:Message UtcTime="2021-07-02T17:00:00.346Z">
         <tt:Source>
          <tt:SimpleItem Name="VideoSoruce" Value="VideoSourceConfig"/>
          <tt:SimpleItem Name="Rule" Value="AreaCount_Rule8"/>
         </tt:Source>
          <tt:Data>
           <tt:SimpleItem Name="AlarmStatus" Value="false"/>
           <tt:SimpleItem Name="Count" Value="1"/>
           <tt:SimpleItem Name="AlarmCount" Value="5"/>
           <tt:SimpleItem Name="AlarmTime" Value="PT10S"/>
         </tt:Data>
        </tt:Message>
      </wsnt:Message>
  </wsnt:NotificationMessage>
 </tt:Event>
</tt:MetaDataStream>
```

In case of sending when the alarm is activated.

```
<?xml version="1.0" encoding="UTF-8"?>
<tt:MetaDataStream xmlns:tt="http://www.onvif.org/ver10/schema"
   xmlns:wsnt="http://docs.oasis-open.org/wsn/b-2"
 <tt:Event>
   <wsnt:NotificationMessage>
     <wsnt:Topic Dialect="http://www.onvif.org/ver10/tev/topicExpression/ConcreteSet"
                  xmlns:tns1=http://www.onvif.org/ver10/topics
                  xmlns:tnspana1=”http://panasonic.co.jp/sn/psn/2010/event/topics”>
       tns1:RuleEngine/CountAggregation/tnspana1:AreaCounter
     </wsnt:Topic>
      <wsnt:Message>
       <tt:Message UtcTime="2021-07-02T16:53:00.256Z">
          <tt:Source>
            <tt:SimpleItem Name="VideoSoruce" Value="VideoSourceConfig"/>
            <tt:SimpleItem Name="Rule" Value="AreaCount_Rule2"/>
          </tt:Source>
          <tt:Data>
            <tt:SimpleItem Name="AlarmStatus" Value="true"/>
            <tt:SimpleItem Name="Count" Value="10"/>
            <tt:SimpleItem Name="AlarmCount" Value="5"/>
            <tt:SimpleItem Name="AlarmTime" Value="PT10S"/>
          </tt:Data>
        </tt:Message>
      </wsnt:Message>
   </wsnt:NotificationMessage>
 </tt:Event>
</tt:MetaDataStream>
```

6. TCP alarm notification

The specification of the TCP alarm notification Protocol is described in the following document.

Command_interface_Panasonic_H.265models_verx.xx.pdf
7.11. Panasonic Alarm Protocol(TCP notification)

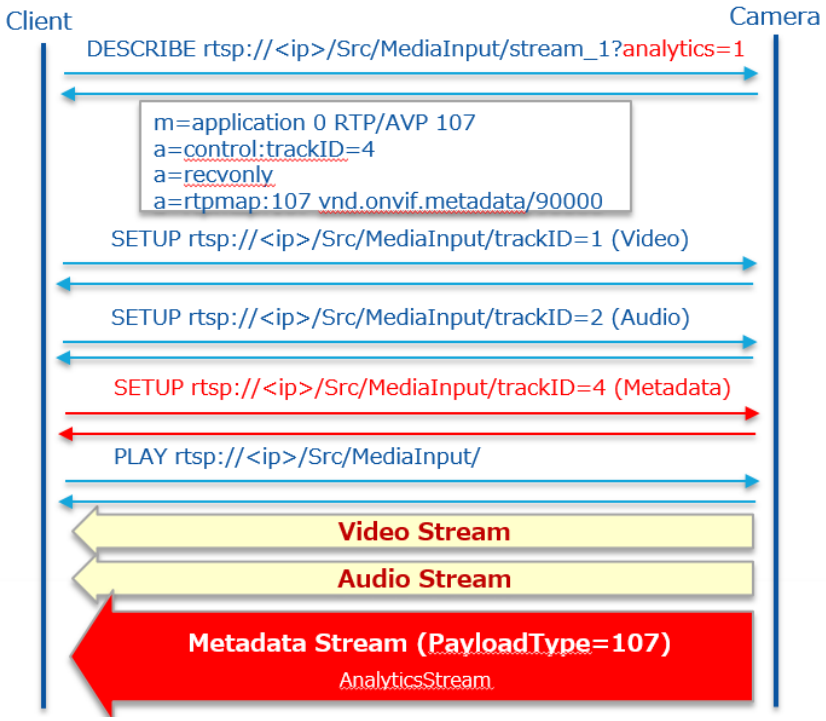
TCP alarm notification of AI-VMD and Occupancy detection is send by the following message ID.

Message name	Extension area		
	Category	Message ID	Message(ASCII)
AI-VMD			
INTRUDER ALARM	0x01	0x32	INTRUDER ALARM HUMAN INTRUDER ALARM VEHICLE
LOITERING ALARM	0x01	0x33	LOITERING ALARM HUMAN LOITERING ALARM VEHICLE
DIRECTION ALARM	0x01	0x34	DIRECTION ALARM HUMAN DIRECTION ALARM VEHICLE
Occupancy detection			
Occupancy Alarm (Area1)	0x01	0x62	OCCUPANCY ALARM(AREA1)
Occupancy Alarm (Area2)	0x01	0x63	OCCUPANCY ALARM(AREA2)
Occupancy Alarm (Area3)	0x01	0x64	OCCUPANCY ALARM(AREA3)
Occupancy Alarm (Area4)	0x01	0x65	OCCUPANCY ALARM(AREA4)
Occupancy Alarm (Area5)	0x01	0x66	OCCUPANCY ALARM(AREA5)
Occupancy Alarm (Area6)	0x01	0x67	OCCUPANCY ALARM(AREA6)
Occupancy Alarm (Area7)	0x01	0x68	OCCUPANCY ALARM(AREA7)
Occupancy Alarm (Area8)	0x01	0x69	OCCUPANCY ALARM(AREA8)

7. Appendix

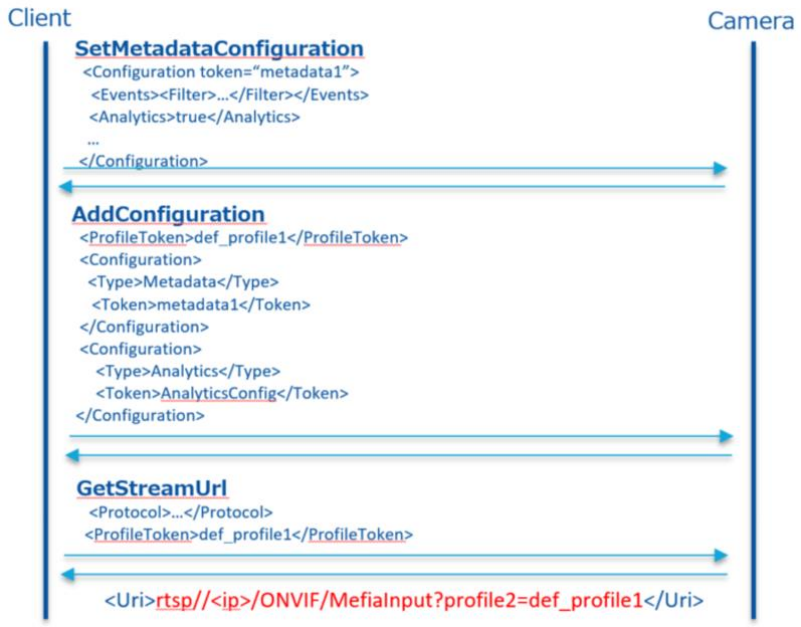
7.1. Transmission sequence of ONVIF metastream

[RTSP URL]
*Send request(RTSP URL) with “analytics=1” in case of requesting Analytics stream.
*Send request with “event=1” in case of requesting Event stream.
*Send request with “analytics=1&event=1” in case of requesting Analytics stream and Event stream.



[ONVIF]
* Configure by ONVIF commands
- SetMetadataConfiguration(Event filter, analytics flag)
- AddConfiguration(Add “metadata1” and “AnalyticsConfig” at “MediaProfile”)

* Get RTSP URL by ONVIF commands(GetStreamUrl)



* Streaming by URL got by ONVIF commands
  - Event stream is also sent by streaming analytics