AI-VMD

External Interface Specification

V1.00

i-PRO Co., Ltd.

Revision history

Ver.	Date	Item no.	Comment	Revision trigger
1.00	2022/7	All	First edition	—

Table of contents

1. Abstract

This is the specification for the external interface for AI-VMD, which provides the following 2 functions.

AI motion detection

This function provides the following 4 operation modes to detect moving objects and issue an alarm accordingly.

- Intruder detection: Detect moving objects intruding the specified area.
- Loitering detection: Detect moving objects loitering within a specified area for a specified time length.
- Direction detection: Detect moving objects that is crossing the specified area towards a specified direction.
- Cross Line detection: Detect moving objects that crossed a specified line towards a specified direction.

In addition, the application allows you to identify whether the moving object is a person, vehicle, or bicycle.

Count function

This function(referred to below as Cross Line Counting) is that the camera counts the number of people, vehicle and bike which crossed the lines and stores the counting result to internal memory as CSV files. 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 count will be reset by "Count data storage interval " setting.

1.1. Output interface

The output interface for AI-VMD is listed below.

Function	Output I/F							
	CGI (Get the count result)	CGI (get_io2)	Stream Additional information	ONVIF (Analytics Stream)	ONVIF (Event Stream)	TCP alarm notification	HTTP POST	MQTT
AI motion detection	—	○	○	○	○	○	—	—
Cross Line Counting	○	—	○	○	○	—	○	○

2. CGI command interface

2.1. Check application status

The following steps allow you to check whether or not the application is already installed in your camera.

① Enter the following URL while your camera is running.

http:// (Camera IP address) /cgi-bin/getinfo?FILE=1

② The message below indicates whether AI-VMD is installed or not.

EXTAPP1= AI-VMD EXTAPP2= EXTAPP3=

“AI-VMD” may appear in the place of EXTAPP2 or EXTAPP3 above depending on the order of the installation.

2.2. Set up CGI specification

2.2.1. Structure

[Command interface]

Method: POST

[CGI]

・Multi-sensor camera (The example below is in the case of specifying channel 1)

http:// (Camera IP address)/cgi-bin/adam.cgi?methodName=sendDataToAdamApplication&appName=iVmdApp&channel=1&s_appDataType=0&s_appData= (base64 data)

・All cameras except for muti-sensor camera

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

CGI URL：http:// (Camera IP address)/cgi-bin/adam.cgi

Name of API: sendDataToAdamApplication

[Request Parameters]

Parameter	Description
appName	「iVmdApp」(fixed)
channel	Channel details of multi-sensor camera ∗Given only to multi-sensor camera
s_appDataType	Data type transmitted. Always “0” if this is a CGI used for the setup
s_appData	Base 64 encoded preference data

[Base64 Data]

Parameter name	Description
appMethod	Always “set_data” when this is a CGI used for the setup
Parameter preferences	Please refer to section 9.1

2.2.2. Transmission procedure

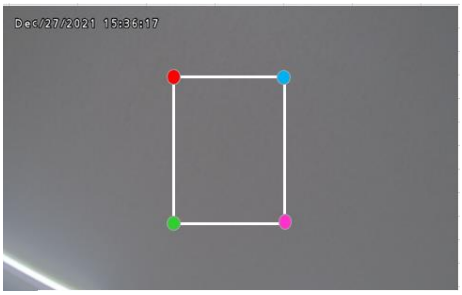
① Create preference data (before base64 encoding) that follows “s_appData=”

Use JSON format as shown below to create the preference data. Refer to section 9.1 for the parameters (key, value).

{{"key":"value"}}

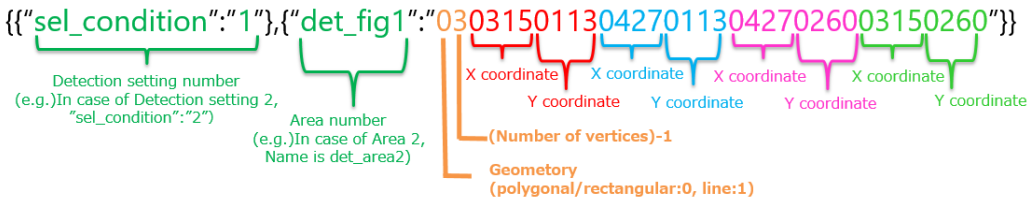
The rule of parameter format is described as follows.

■**Detection area coordinate**



【A sample detection area coordinates】

Example: The coordinates of the 4 vertexes of the specified detection area 1 (315,113), (427,113), (427,260), (315, 260) should be included in the parameter as indicated below.



② Apply base 64 encode to the preference value that follows "s_appData=" (The following is used for all cameras except for multi-sensor camera)

■**Before Base64 encoding**

http:// (Camera IP address)/cgi-bin/adam.cgi?methodName=sendDataToAdamApplication&appName=iVmdApp&s_appDataType=0&s_appData={{sel_condition:1},{det_fig1:03031501130427011304270260031502600}}

■**After Base64 encoding**

http:// (Camera IP address)/cgi-bin/adam.cgi?methodName=sendDataToAdamApplication&appName=iVmdApp&s_appDataType=0&s_appData=e3tzZWxfY29uZGl0aW9uOjF9LHtkZXRfZmlnMTowMzAzMTUwMTEzMDQyNzAxMTMwNDI3MDI2MDAzMTUwMjYwfX0=

③ Send the preference data after base64 encoding using CGI

2.3. Retrieve preference data using CGI

You can retrieve the previously set preference data of your application by using CGI below.

・For multi-sensor camera (Example below is in the case of specifying channel 1)

http:///(Camera IP address)/cgi-bin/adam.cgi?methodName=getApplicationPreference&appName=iVmdApp&channel=1

・All cameras except for muti-sensor camera

http:///(Camera IP address)/cgi-bin/adam.cgi?methodName=getApplicationPreference&appName=iVmdApp

[Request Parameters]

Parameter name	Description
appName	Always「iVmdApp」
channel	Channel information of the multi-sensor camera ∗Given only to multi-sensor camera

[Normal Response Parameters]

Parameter name		Description
funcId		FuncID
preferenceVersion		Preference data version
preference		Information of preference data
	prefName	Name of preference data
	prefType	Type of preference data(Boolean / Integer / String / Enumeration / Binary) Boolean: Logical Types Integer: 64-bit signed integer type String: String type Enumeration: Enums Binary: Binary Types
	enumerationList	List of Enumeration ∗Given only when prefName==Enumeration
	defaultValue	Default value This is expressed by prefType as follows.

		In the case of prefType == Boolean: TRUE or FALSE In the case of prefType == Integer: Signed 64-bit decimal integer values In the case of prefType == String: String In the case of prefType == Enumeration: One of the enumerationList In the case of prefType == Binary: Base64 encoded data of binary
	webApiAccess	Permission to access from WebAPI ReadWrite: Possible to read and write from WebAPI Read: Only read from WebAPI (Impossible to Set)
	value	Current value The method of expression by the difference of prefType is the same as defaultVaule.

[Abnormal Response Parameters]

Parameter name	Description
faultCode	Error Code
faultString	Error String

HTTP response header = "400" Bad Request

faultCode="1" faultString="Invalid Parameter"

Wrong parameter

faultCode="2" faultString="Invalid Method"

Not implemented method name

faultCode="3" faultString="Invalid Install ID"

Invalid Install ID

faultCode="4" faultString="Invalid Process ID"

Invalid Process ID

faultCode="5" faultString="Invalid Exec ID"

Invalid Execution ID

faultCode="6" faultString="Invalid Registration Key"

Invalid Registration Key

faultCode="7" faultString="Invalid Cipher Control"

Decription failure

faultCode="8" faultString="Can't Execute Script File"

Can't execute script file

faultCode="9" faultString="Invalid Application Package"

Invalid Application Package

faultCode="10" faultString="Invalid Protocol"
Invalid requested protocol
faultCode="20" faultString="Not Supported"
Not support API

HTTP response header = "403" Forbidden
faultCode="11" faultString="Permission denied"
Don't have exectution right
faultCode="12" faultString="Registration Key Expired"
Registration Key Expired

HTTP response header = "409" Conflict
faultCode="13" faultString="Bad Application Status"
Application status doesn't expect the requested function

HTTP response header = "500" Internal Server Error
faultCode="14" faultString="File Access Error"
Internal error (File access error)
faultCode="15" faultString="I/O error"
Internal error (device I/O error)
faultCode="16" faultString="Not Enough Memory"
Internal error (Not Enough Memory)
faultCode="17" faultString="Application Start Error"
Internal error (Application Start)
faultCode="18" faultString="Internal Error"
Internal error (Other internal error)

[Response Format (Normal) : JSON]

Refer to section 9.1 for the parameters used.

```
{ "funcId":"FUNCID",
  "preferenceVersion":"Preference data version",
  "preference":[
        {"prefName":"Name of preference data",
        "prefType":"Type of preference data",
        "enumerationList":["ENUM value 1","ENUM value 2","ENUM value 3",..],
        "defaultValue":"Default value",
```

"webApiAcess":"Permission to access from WebAPI",
"value":"Current value"},
{...},
...
]
}

[Response Format (Abnormal) : JSON]

{"faultCode":"Error Code","faultString":"Error String"}

2.4. 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

2.4.1. 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,
 "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]]},
   ],
“Line1_cntobj“:[“xxxx“],
“Line2_cntobj“:[“xxxx“],
“Line3_cntobj“:[“xxxx“],
~
“Line8_cntobj“:[“xxxx“]
}
```

【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
Line1.list Line2.list Line3.list Line4.list Line5.list Line6.list Line7.list Line8.list	[“Date and time(UTC)”, Number of detected objects direction “In”(0~100),Number of detected objects direction “Out”(0~100)]		Number of objects 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 objects]: Number of objects detected for 1 minute.
Line1_cntobj Line2_cntobj Line3_cntobj Line4_cntobj Line5_cntobj Line6_cntobj Line7_cntobj Line8_cntobj	Human, Vehicle, Bike	String	Target object to be counted * If the detection line is not set, send it blank. e.g.) If human is the target of counting, it will be as follows. "Line1_cntobj":["Human"]

* It does not include information on the time when the detection 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)

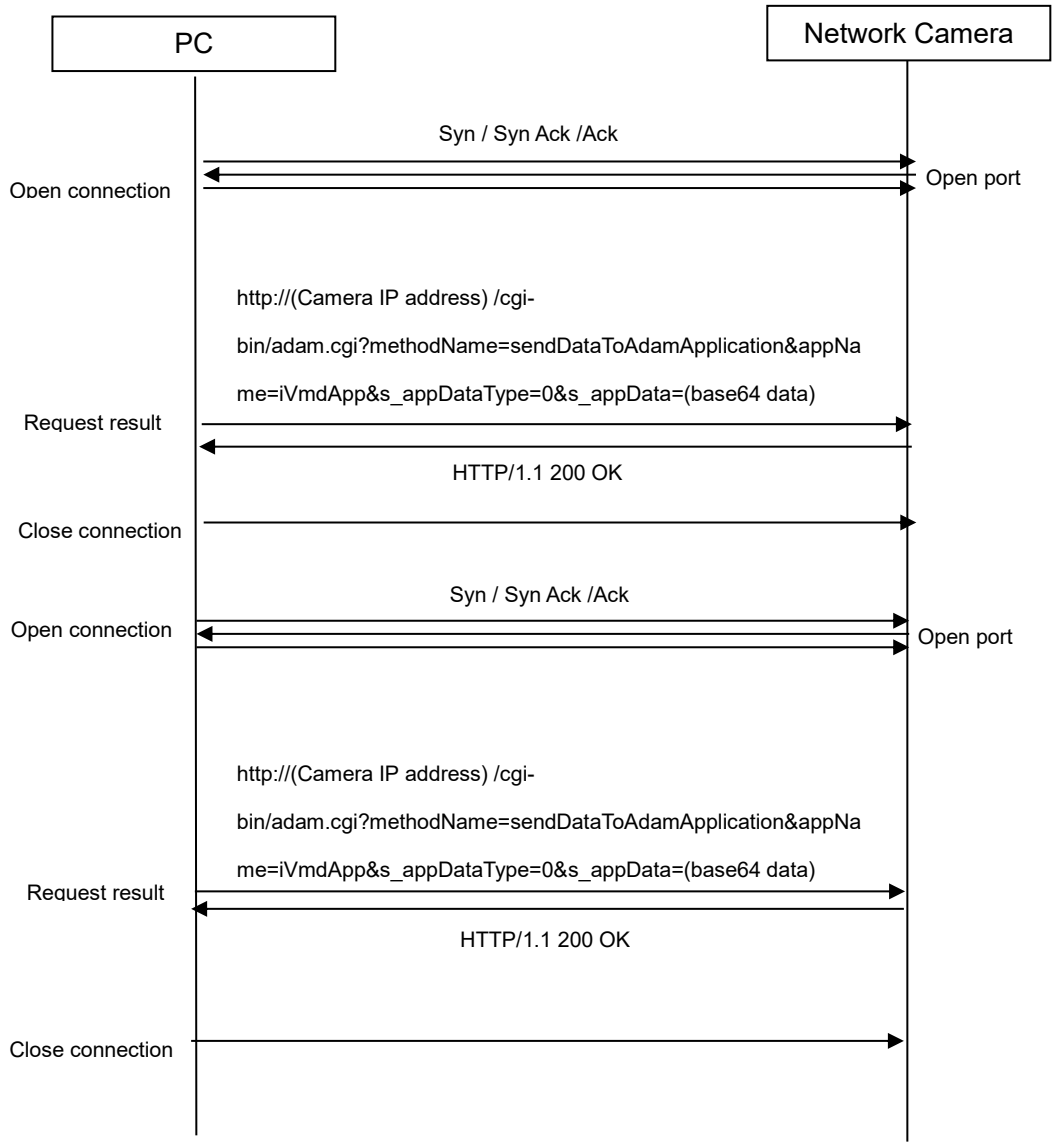
faultCode="15" faultString="I/O error"
Internal error (I/O error)

faultCode="16" faultString="Not Enough Memory"
Internal error (insufficient memory)

faultCode="18" faultString="Internal Error"
Internal error (other error)

2.4.2. Getting JSON file

Sequence



Response format

e.g.) Multi-Sensor Camera(MAC address = 00:80:45:0d:00:01, channel number = 1), min = {10}, detection line = {line 1, 2}, effective detection line = {line 1, 2}, target object ={line 1: Human, line 2: Vehicle}, Transmission time: {JST 2021/1/11 18:10:00}


```
{
   “CameraIPaddress”:“192.168.0.10”,
   "CameraMACaddress":"00:80:45:0d:00:01",
   "Ch":"1",
   ”Time":”2021/1/11 9:10:00”,
   "TimeZone":"+0900",
   "SummerTime":0,
    “Line1”:[
        {“list”: [
                        [“2021/1/11 9:00”, 7, 6],
                        [“2021/1/11 9:01”, 7, 7],
                        [“2021/1/11 9:02”, 8, 10],
                        [“2021/1/11 9:03”, 9, 11],
                        [“2021/1/11 9:04”, 6, 5],
                        ["2021/1/11 9:05", 5, 5],
                        ["2021/1/11 9:06", 6, 7],
                        ["2021/1/11 9:07", 6, 8],
                        ["2021/1/11 9:08", 7, 6],
                        ["2021/1/11 9:09", 8, 7]
                    ]
      },
     ],
       “Line2”:[
        {“list”: [
                        [“2021/1/11 9:00”, 4, 5],
                        [“2021/1/11 9:01”, 5, 6],
                        [“2021/1/11 9:02”, 12, 11],
                        [“2021/1/11 9:03”, 12, 10],
                        [“2021/1/11 9:04”, 9, 8],
                        ["2021/1/11 9:05", 10, 8],
                        ["2021/1/11 9:06", 10, 10],
                        ["2021/1/11 9:07", 16, 15],
                        ["2021/1/11 9:08", 17, 17],
                        ["2021/1/11 9:09", 18, 17]
        },
     ],
    “Line3”:[
        {“list”: []},
    ],
                                        ~
  “Line8”:[
        {“list”: []},
   ],
  “Line1_cntobj“:[“Human“],
  “Line2_cntobj“:[“Vehicle“],
  “Line3_cntobj“:[],
                                        ~
  “Line8_cntobj“:[]
}
```

2.5. 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_info	Kind of the csv data movcnt_info:Line cross count result *This parameter can't be omitted
mode	range multi latest	Kind of response range: Get storing period multi: Get csv files by a date. latest: Get latest csv file *This parameter can't be omitted
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_info},{mode:range}}

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

Getting the Line cross count result 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_info},{mode:multi},{year:2021},{month:7},{date:29},{days:5},{hour:16}}

*{{appMethod:csv},{kind:movcnt_info},{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}_info,{mode:multi},{year:2021},{month:7},{date:30},{days:5},{hour:5}}

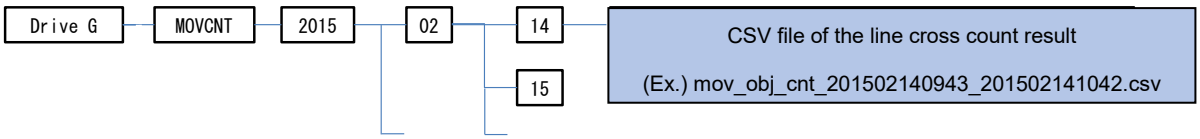
*{{appMethod:csv},{kind:movcnt_info},{mode:multi},{year:2021},{month:7},{date:30},{days:5},{hour:5}} is base64 data.

Getting the latest 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_info},{mode:latest}}

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

2.5.1. Directory structure and file name

CSV files are stored in the internal memory of the camera. The content of internal memory can be seen as a drive G.



2.5.2. CSV file format

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 objects 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 objects 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 count results e.g.) In case of 15min-> 00:15
timezone	-12:00 to +12:00 (6 columns)	timezone
summertime	IN,	summertime

	OUT	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 objects. Direction “In” (Line1)
count_out1	0 to 65535	Counting result of the number of objects. Direction “Out” (Line1)

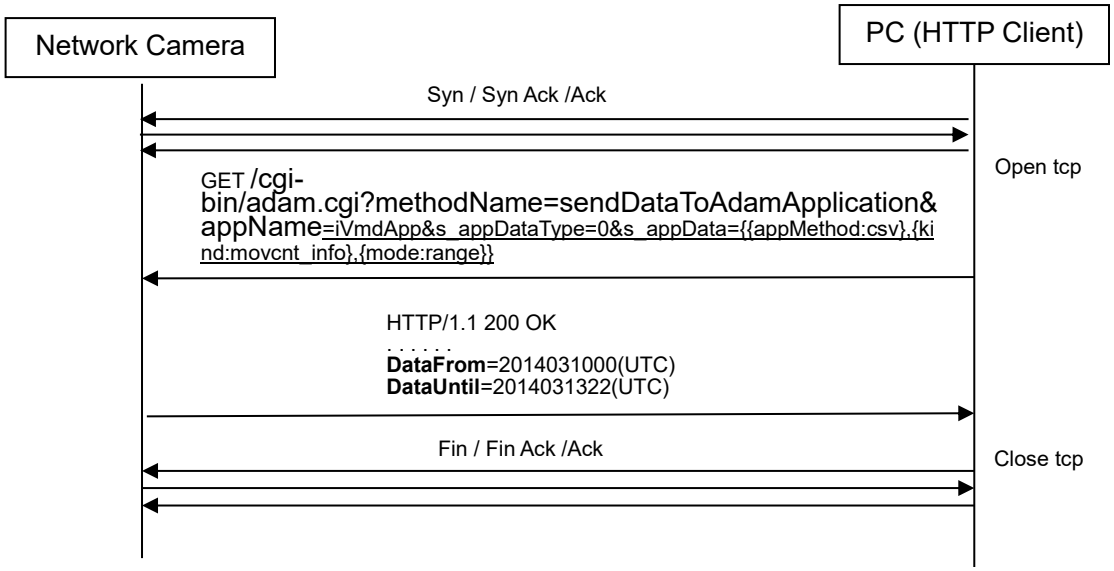
* The above descriptions are common in other lines(s_x2,s_y2,,,, e_x12,e_y12).
* When a line wasn't set, both coordinates information set to (0, 0).

2.5.3. Getting the CSV file

The command sequence to acquire the csv files is described in this chapter.

2.5.3.1. Get storing period (mode:range)

Sequence



Response format

HTTP1.1 200OK [CR][LF]

Status 200[CR][LF]

. . . .

Content-Length: xxxxx[CR][LF]

DataFrom=YYYYMMDDHHmm(UTC)[CR][LF]

DataUntil=YYYYMMDDHHmm(UTC)[CR][LF]

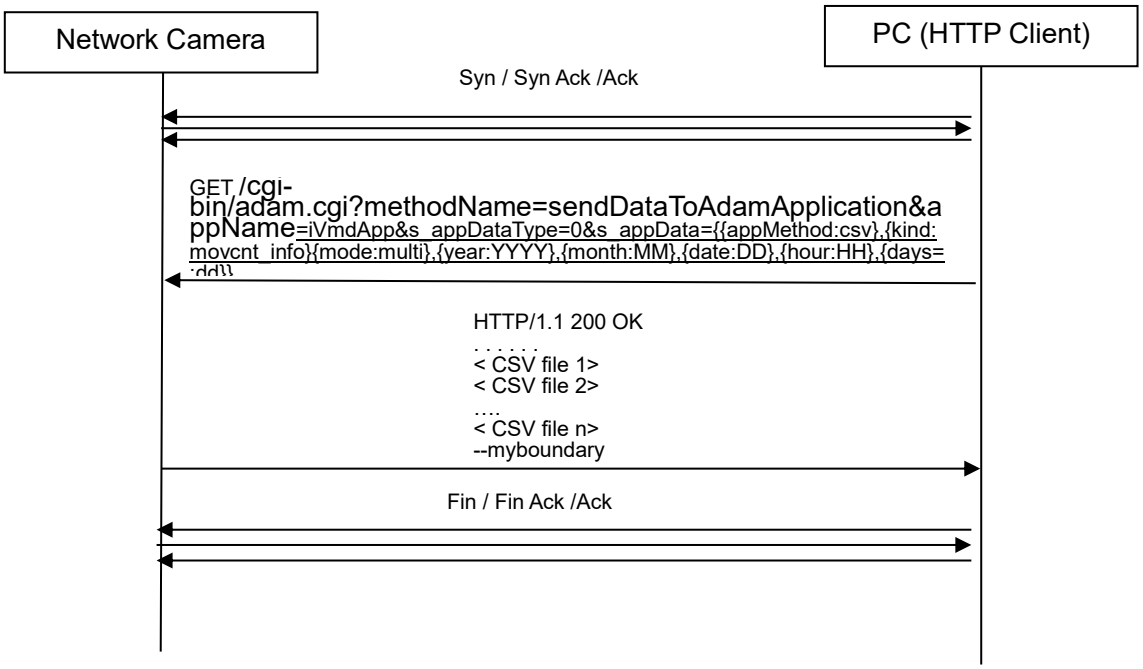
[CR][LF]

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)

2.5.3.2. Get csv file by a date (mode:multi)

Sequence

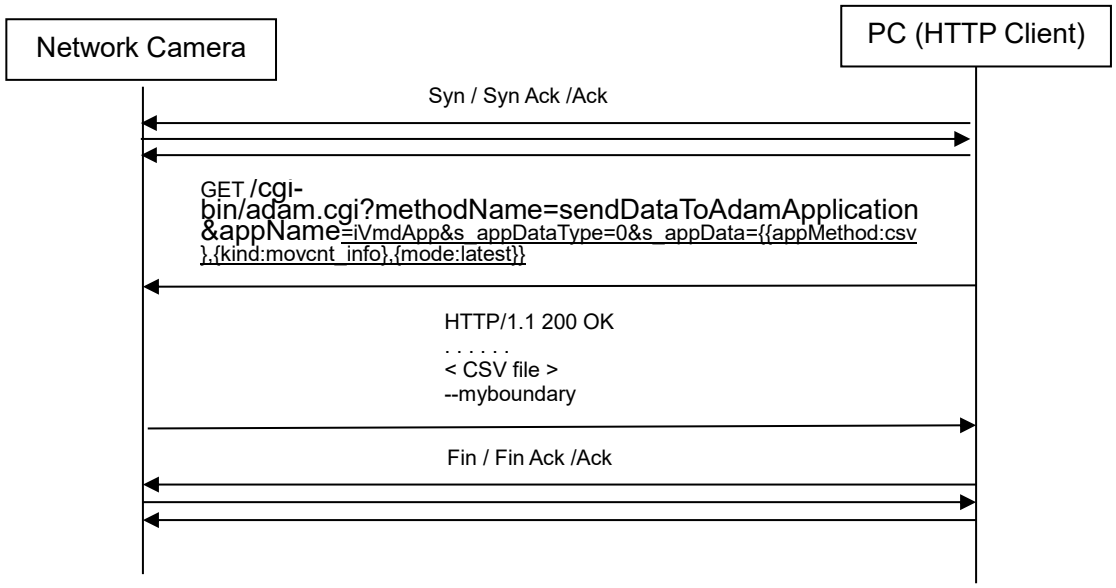


Response format

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]

2.5.3.3.Get latest csv file (mode:latest)

Sequence



Response format

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** >

2.5.3.4. Error response

Error response

HTTP1.1 200OK [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).

2.6. get_io2

You can review alarm data such as whether or not alarm has been issued, area number of the issued alarm and the target object that triggered the alarm by transmitting the following CGI.

[CGI]

http:// (Camera IP address)/cgi-bin/get_io2?monitor&format=3

[Response parameter]

Parameter (*1)	**Status**	**Display**
intruder intruder_ch2 intruder_ch3 intruder_ch4	No intrusion detected	intruder=False
	Intrusion detected	intruder=True
loitering loitering_ch2 loitering_ch3 loitering_ch4	No loitering detected	loitering=False
	Loitering detected	loitering=True

direction direction_ch2 direction_ch3 direction_ch4	No direction detected	direction=False
	Direction detected	direction=True
crossline crossline_ch2 crossline_ch3 crossline_ch4	No cross line detected	crossline=False
	Cross line detected	crossline=True
intruder_area intruder_area_ch2 intruder_area_ch3 intruder_area_ch4	No intruder detected	intruder_area=0000
	An intruder detected	intruder_area=XXYY *2
loitering_area loitering_area_ch2 loitering_area_ch3 loitering_area_ch4	No loitering detected	loitering_area=0000
	Loitering detected	loitering_area=XXYY *2
direction_area direction_area_ch2 direction_area_ch3 direction_area_ch4	No direction detected	direction_area=0000
	Direction detected	direction_area=XXYY *2
crossline_area crossline_area_ch2 crossline_area_ch3 crossline_area_ch4	No cross line detected	crossline_area=0000
	Cross line detected	crossline_area=XXYY *2
alarm_object alarm_object_ch2 alarm_object_ch3 alarm_object_ch4	ivmd no alarm	alarm_object=False
	Intrusion detection subject: people	alarm_object=INTRUDER ALARM HUMAN
	Intrusion detection subject: : vehicle	alarm_object=INTRUDER ALARM VEHICLE
	Intrusion detection subject: : bicycle	alarm_object=INTRUDER ALARM BICYCLE
	Loitering detection subject: : people	alarm_object=LOITERING ALARM HUMAN
	Loitering detection subject: : vehicle	alarm_object=LOITERING ALARM VEHICLE
	Loitering detection subject: :	alarm_object=LOITERING ALARM BICYCLE

	bicycle	
	Intrusion detection subject: : person	alarm_object=DIRECTION ALARM HUMAN
	Intrusion detection subject: : vehicle	alarm_object=DIRECTION ALARM VEHICLE
	Intrusion detection subject: bicycle	alarm_object=DIRECTION ALARM BICYCLE
	Cross line detection subject: : people	alarm_object=CROSS LINE ALARM HUMAN
	Cross line detection subject: : vehicle	alarm_object=CROSS LINE ALARM VEHICLE
	Cross line detection subject: : bicycle	alarm_object=CROSS LINE ALARM BICYCLE

*1 Parameters whose the end of the parameter name is “_ch2~4” are given only to multi-sensor camera. If the name is "_ch2", it is the detection information of camera 2, if it is "_ch3", it is the detection information of camera 3, and if it is "_ch4", it is the detection information of camera 4.

*2 You can include detection area 01 to FF in XX in the first half or in YY in the latter half by turning on the setting of alarm area data　(This is applicable only for v3.00 or later of this software)

[Setting CGI]
http:// (Camera IP address)/cgi-bin/pana_alm?ivmd_ext=1

Detection area/line1・・・01
Detection area/line2・・・02
Detection area/line3・・・04
Detection area/line4・・・08
Detection area/line5・・・10
Detection area/line6・・・20
Detection area/line7・・・40
Detection area/line8・・・80
When detection covers more than one area, it is OR value of the numbers above.

For example, it is displayed as follows if the detection area 2 and 3 both detected objects that is set as an intrusion detection as a detection condition 01.

intruder_area=0106

3. Additional information

Meta data is given to H.264/H.265 RTP header and JPEG header when this feature is in use. Refer to the document below for the details of meta data.

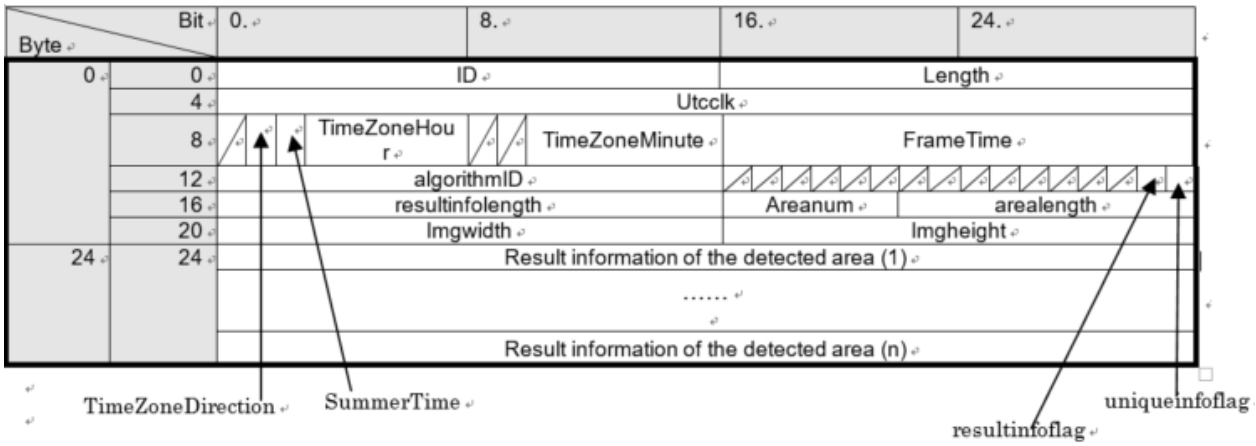
Command_interface_Panasonic_H.265models_verx.xx.pdf

13.7 Location of meta information in each stream

Prior to use the feature that provides metadata, set “Additional information type” to “With detected object information (alarm frame information)”. [ivmd_info_type=1]

3.1. AI motion detection

3.1.1. Basic information



[Data layout]

[List of the basic data]

Parameter	Length (Bit)	Values and comments
ID	16	0x002F (Fixed ID for AI-VMD data)
Length	16	Length of AI-VMD data (including ID and Length) in byte
Utcclk	32	The career second from 1970
TimeZoneDirection	1	Time zone (±) 0x00 : ＋

		0x01 : ー
SummerTime	1	0x00 : Outside of daylight saving time 0x01 : During daylight saving time
TimeZoneHour	5	Time zone (hour) 0x00: 0 hour, 0x01: 1 hour, 0x02: 2 hours, 0x03: 3 hours 0x04: 4 hours, 0x05: 5 hours, 0x06: 6 hours, 0x07: 7 hours 0x08: 8 hours, 0x09: 9 hours, 0x0a: 10 hours, 0x0b: 11 hours 0x0c: 12 hours, 0x0d: 13 hours, 0x0e: 14 hours, 0x0f: 15 hours 0x10: 16 hours, 0x11: 17 hours, 0x12: 18 hours, 0x13: 19 hours 0x14: 20 hours, 0x15: 21 hours, 0x16: 22 hours, 0x17: 23 hours
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	0x0100 (fixed)
resultinfoflag	1	Result information flag 0 (b): Not include the result information 1 (b): Include the unique information
uniqueinfoflag	1	0 (b) (Fixed value)
resultinfolength	16	Length of the result information in byte
Areanum	6	The number of detected frames
arealength	10	The data length of a detection result in byte
Imgwidth	16	Width of the image for the AI-VMD detection
Imgheight	16	Height of the image for the AI-VMD detection

3.1.2. Result information

Result information of each detected frame

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

[Data layout]

[List of result data]

Parameter	Length (Bit)	Values and comments
areaID	16	ID of the detected frame 0 ～ 65535
Dtctarea	16	0x0001：Detection program1-Detection area1 0x0002：Detection program1-Detection area2 0x0004：Detection program1-Detection area3 0x0008：Detection program1-Detection area4 0x0010：Detection program1-Detection area5 0x0020：Detection program1-Detection area6 0x0040：Detection program1-Detection area7 0x0080：Detection program1-Detection area8 0x0100：Detection program2-Detection area1 0x0200：Detection program2-Detection area2 0x0400：Detection program2-Detection area3 0x0800：Detection program2-Detection area4 0x1000：Detection program2-Detection area5 0x2000：Detection program2-Detection area6 0x4000：Detection program2-Detection area7 0x8000：Detection program2-Detection area8
Almtype	4	Alarm status 0x01: Intruder detection 0x02: Loitering detection 0x03: Direction detection

		0x05: Cross line detection 0x08: AI detection 0x0F: Not alarmed * This value is 0x08 only when “Additional information type” is set to “With detected object information (AI frame information)”. [ivmd_info_type=2]
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：A→B 0x0a：B→A 0x0b：A⇔B 0x00：Not alarmed for direction/line cross detection
almobj	8	Detected object at the time of alarm issue 0x01: human 0x02: vehicle 0x03: bicycle
Hstart	16	X coordinate (Upper left) of the detection frame ∗1
Vstart	16	Y coordinate (Upper left) of the detection frame ∗1
Hcnt	16	Width of the detection frame ∗1
Vcnt	16	Height of the detection frame ∗1

AI-VMD information is refreshed every 100 milliseconds.

*1 Overall resolution depends on the settings of your camera

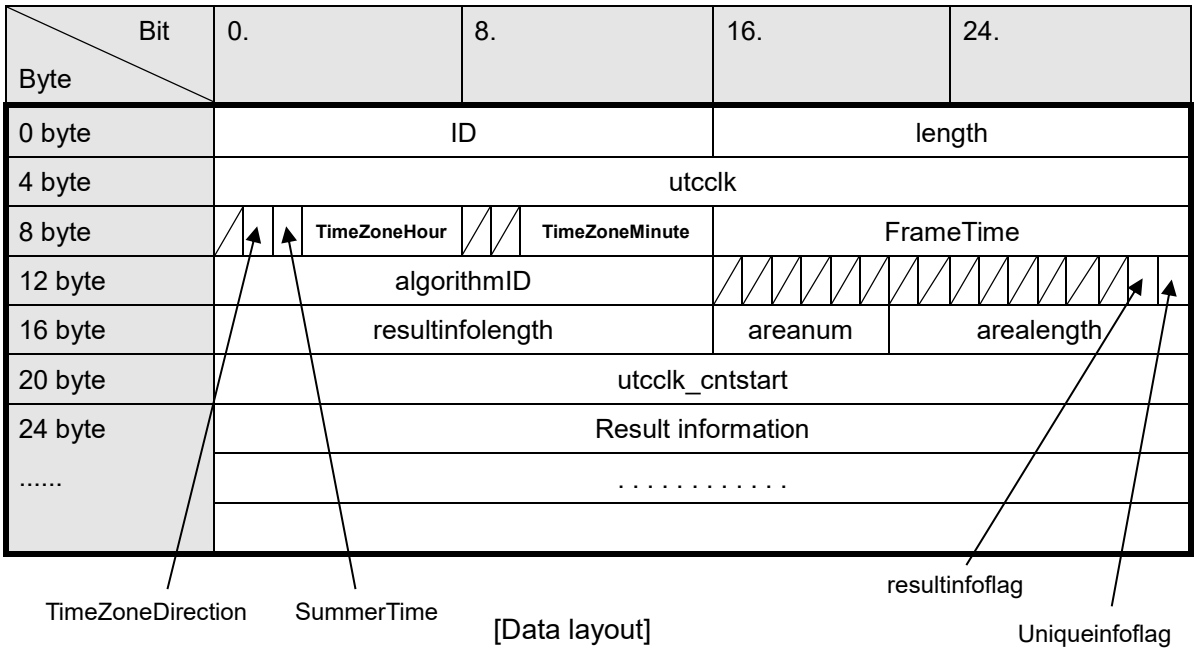
800×450 when the image capture mode is 16:9 and image rotation is 0° /180°

450×800 when the image capture mode is 16:9 and image rotation is 90° /270°

800×600 when the image capture mode is 4:3 and image rotation is 0° /180°

3.2. Cross Line Counting

3.2.1. Basic information



[Data layout]

[List of the basic data]

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 career seconds from 1970(UTC time).
Result information	Variable	Count information

3.2.2. Result information

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

[Data layout]

[List of result data]

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)
cntobj	3	Target object to be counted 00: Not count 01: Vehicle 02: Bike 04: Human
countIn	16	The number of counting objects (direction: In)
countOut	16	The number of counting objects (direction: Out)

4. ONVIF Meta Stream

There are two kinds of metadata information

① Analytics stream：Periodically transmit detection frame data. The transmission interval is as follows.

-All cameras except for multi-sensor camera:

(when the image capture mode is 30fps)10fps

(when the image capture mode is 25fps) 8.3fps

-Multi-sensor camera:

(when the image capture mode is 15fps/30fps) 3.75fps

(when the image capture mode is 12.5fps/25fps) 3.1fps

② Event stream: Transmitted when event occurs. Send interval is different for each function.

4.1. Analytics Stream

4.1.1. 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

4.1.2. Meta Steam format sample

```
<?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>
```

4.2. Event stream

The format of Event stream is different for each function of AI-VMD.

4.2.1. AI motion detection

Event	Parameter	Value	Comment
Intrusion detection	UtcTime	Date and Time (UTC)	Year, date and time of metadata transmission
	VideoSource	VideoSourceConfig	Token name of VideoSourceConfiguration
	Rule	Rule1, Rule2, Rule3, Rule4, Rule5, Rule6, Rule7, Rule8	Area number (detection area 1 ~ 8)
	ObjectId	4-Byte integer	ID number of detected object
	IsInside	true, false	true: The object is in the detection area false: The object is outside of the detection area
	ClassTypes	Human, Vehicle, Bicycle	Classification of the object
	Image	Base64 encoded	JPEG image at the time of the event
Loitering detection	UtcTime	Date and Time (UTC)	Year, date and time of metadata transmission
	VideoSource	VideoSourceConfig	Token name of VideoSourceConfiguration
	Rule	Rule1, Rule2, Rule3, Rule4, Rule5, Rule6, Rule7, Rule8	Area number (detection area 1 ~ 8)
	ObjectId	4-Byte integer	ID number of detected object
	Since	Date and Time (UTC)	The time from which the object started loitering.
	ClassTypes	Human, Vehicle, Bicycle	Classification of the object
	Image	Base64 encoded	JPEG image at the time of the event

Cross line detection	UtcTime	Date and Time (UTC)	Year, date and time of metadata transmission
	VideoSource	VideoSourceConfig	Token name of VideoSourceConfiguration
	Rule	Rule1, Rule2, Rule3, Rule4, Rule5, Rule6, Rule7, Rule8	Area number (detection area 1 ~ 8)
	ObjectId	4-Byte integer	ID number of detected object
	ClassTypes	Human, Vehicle, Bicycle	Classification of the object
	Image	Base64 encoded	JPEG image at the time of the event
Direction detection	UtcTime	Date and Time (UTC)	Year, date and time of metadata transmission
	VideoSource	VideoSourceConfig	Token name of VideoSourceConfiguration
	Rule	Rule1, Rule2, Rule3, Rule4, Rule5, Rule6, Rule7, Rule8	Area number (detection area 1 ~ 8)
	ObjectId	4-Byte integer	ID number of detected object
	Direction	Up/Upper right/Right/Lower Right/ Lower/Lower left/ Left/Upper left	Detected direction
	ClassTypes	Human, Vehicle, Bicycle	Classification of the object
	Image	Base64 encoded	JPEG image at the time of the event

Meta Stream format sample

[Intrusion 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>
```

[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>
```

[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>
```

[Direction 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/Direc
    tionDetector/Moved</wsnt:Topic>
     <wsnt:Message>
      <tt:Message UtcTime="2022-05-25T00:44:46Z">
       <tt:Source>
        <tt:SimpleItem Name="VideoSource" Value="VideoSourceConfig"/>
        <tt:SimpleItem Name="Rule" Value="Rule1"/>
       </tt:Source>
       <tt:Data>
        <tt:SimpleItem Name="ObjectId" Value="342"/>
        <tt:SimpleItem Name="Direction" Value="Left"/>
        <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>
```

4.2.2. Cross Line Counting

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

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 objects crossed the line to Directoin “In” and Direction “Out”.
RightCount	0~65,535 (Decimal number)	Number of objects crossed the line to Direction “Out”.
LeftCount	0~65,535 (Decimal number)	Number of objects crossed the line to Direction “In”.
UtnRightCount UtnLeftCount	0~65,535 (Decimal number)	Number of objects U-turned the line. UtnRightCount : Crossed to Direction “Out” after crossed to Direction “In”. UtnLeftCount : Crossed to Direction “In” after crossed to Direction “Out”.
CountObjHuman	true/false	true: Count human false: Not count human
CountObjVehicle	true/false	true: Count vehicle false: Not count vehicle
CountObjBike	true/false	true: Count bike false: Not count bike

Meta Stream format sample

```
<?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:SimpleItem Name="CountObjHuman" Value="true"/>
               <tt:SimpleItem Name="CountObjVehicle" Value="false"/>
               <tt:SimpleItem Name="CountObjBike" Value="false"/>
          </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:SimpleItem Name="CountObjHuman" Value="false"/>
            <tt:SimpleItem Name="CountObjVehicle" Value="true"/>
            <tt:SimpleItem Name="CountObjBike" Value="true"/>
          </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:SimpleItem Name="CountObjHuman" Value="false"/>
            <tt:SimpleItem Name="CountObjVehicle" Value="true"/>
            <tt:SimpleItem Name="CountObjBike" Value="false"/>
          </tt:Data>
        </tt:Message>
      </wsnt:Message>
    </wsnt:NotificationMessage>
  </tt:Event>
</tt:MetaDataStream>
```

5. TCP alarm notification

Refer to the document below to learn about TCP alarm notification.

Command_interface_Panasonic_H.265models_verx.xx.pdf

7.11. Panasonic Alarm Protocol(TCP notification)

The alarm notification is transmitted in the form of message ID listed below.

Message name	Extension area		
	Category	Message ID	Message (ASCII)
INTRUDER ALARM	0x01	0x32	INTRUDER ALARM XX **** (*1) (*2) (eg.) INTRUDER ALARM INTRUDER ALARM 0102 INTRUDER ALARM HUMAN INTRUDER ALARM VEHICLE 0102
LOITERING ALARM	0x01	0x33	LOITERING ALARM XX **** (*1) (*2) (eg.) LOITERING ALARM LOITERING ALARM 0102 LOITERING ALARM HUMAN LOITERING ALARM VEHICLE 0102
DIRECTION ALARM	0x01	0x34	DIRECTION ALARM XX **** (*1) (*2) (eg.) DIRECTION ALARM

			DIRECTION ALARM 0102 DIRECTION ALARM HUMAN DIRECTION ALARM VEHICLE 0102
CROSS LINE ALARM	0x01	0x38	CROSS LINE ALARM XX **** (*1) (*2) (eg.) CROSS LINE ALARM CROSS LINE ALARM 0102 CROSS LINE ALARM HUMAN CROSS LINE ALARM VEHICLE 0102

*1 XX represents detection object as follows.

Human: HUMAN

Vehicle: VEHICLE

Bicycle: BICYCLE

([ivmd_info_type=0] will not be included if the setting for “metadata class” is set to “Not include data of detection object”)

*2 **** can display information of area/line if your setting of “Include camera alarm area information” is ON. (This is applicable only for v3.00 or later of this software)

[CGI used for the setup]

http:// (Camera IP address)/cgi-bin/pana_alm?ivmd_ext=1

The first half of ** is filled with detection condition 01～02

The latter half of ** is filled with detection area/line 01～FF

Detection area/line1・・・01

Detection area/line2・・・02

Detection area/line3・・・04

Detection area/line4・・・08

Detection area/line5・・・10

Detection area/line6・・・20

Detection area/line7・・・40

Detection area/line8・・・80

When detection covers more than one area, it is OR value of the numbers above.

For example, it is displayed as follows if the detection area 2 and 3 both detected objects that is set as an intrusion detection in detection criteria 01.

INTRUDER ALARM 0106

6. HTTP alarm notification

Refer to the document below to learn about HTTP alarm notification.

Command_interface_Panasonic_H.265models_verx.xx.pdf

7.12. HTTP alarm notification

The HTTP alarm notification is transmitted in the form of alternative characters as follows.

Alternative characters within MHttpUrl# parameter	**Value**
%almsrc	41
%almsrc2	Intrusion detection: 32 Loitering detection: 33 Direction detection: 34 Cross line detection: 38

7. HTTP periodic transmission

7.1. Telegraphic protocol specification

When using the telegraphic protocol between the AI-VMD (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.

7.2. 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

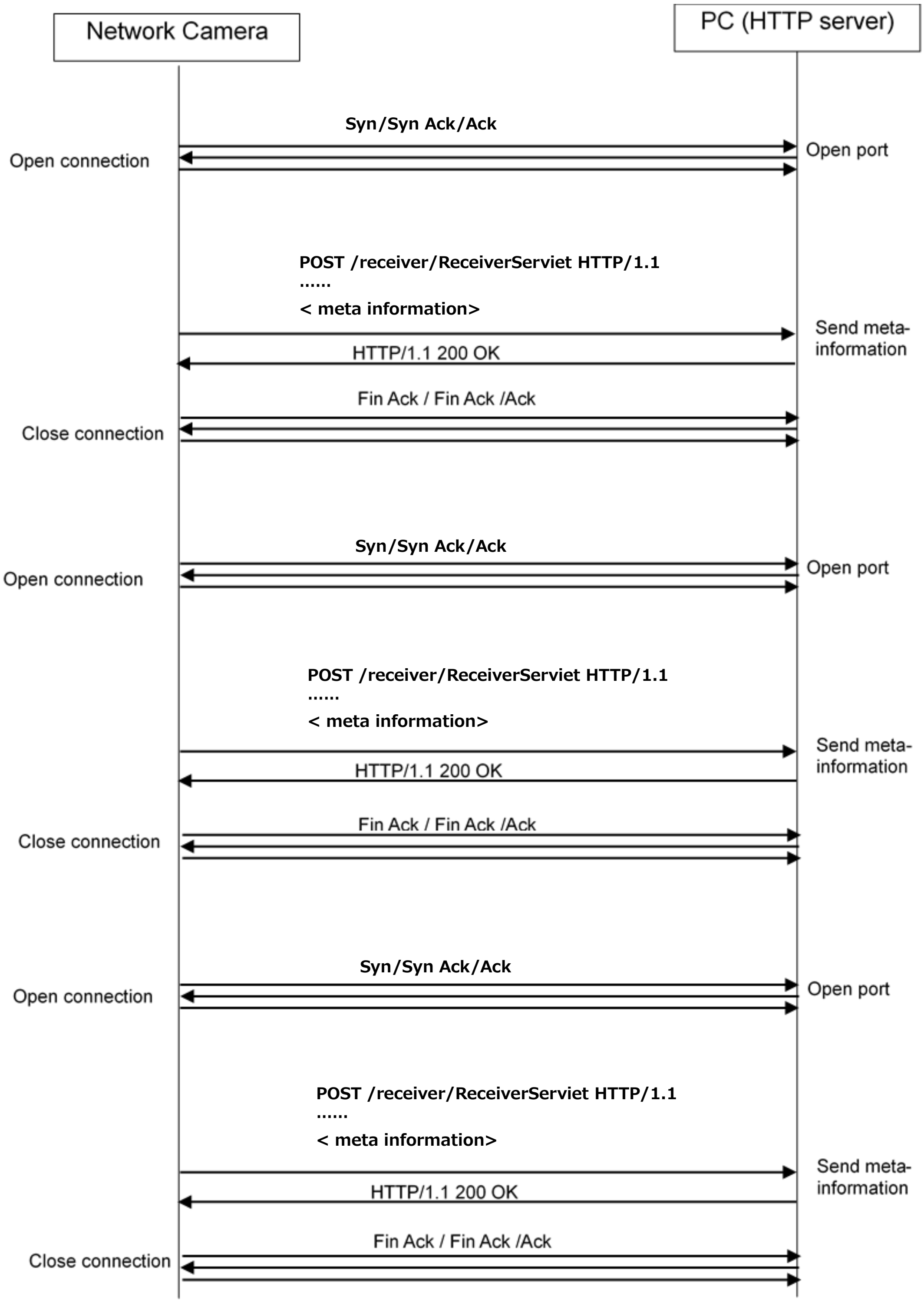
[Count information (Body part)]

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
Line1.list Line2.list Line3.list Line4.list Line5.list Line6.list Line7.list Line8.list	[“Date and time(UTC)”, Number of detected object(Direction “In”), Number of detected object(Direction “Out”)]		Number of objects detected for earch detection line counted by Cross Line Counting.
Line1_cntobj Line2_cntobj Line3_cntobj Line4_cntobj Line5_cntobj Line6_cntobj Line7_cntobj Line8_cntobj	Human, Vehicle, Bike	String	Target object to be counted * If the detection line is not set, send it blank. e.g.) If human is the target of counting, it will be as follows. "Line1_cntobj":["Human"]

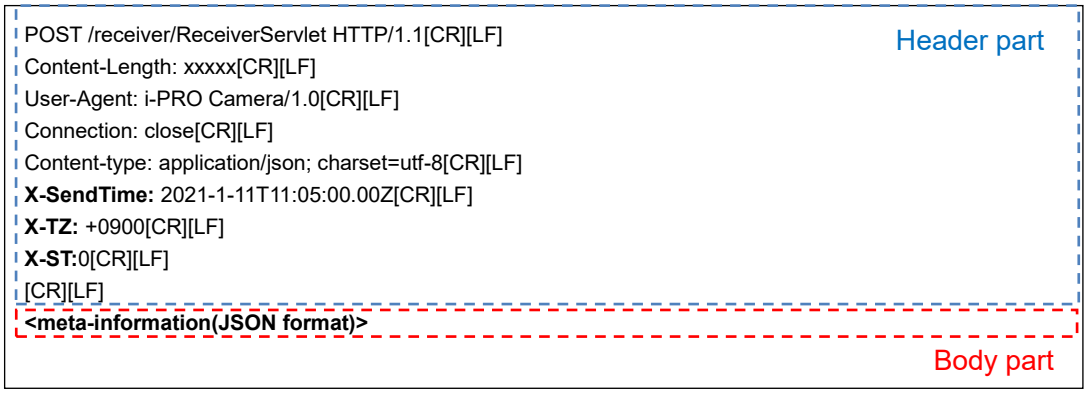
*It does not include information about the time when the detection area and line were not set or were disabled.

7.3. Telegraphic protocol sequence



7.4. Transmission format

An example of the transmission format is shown below.



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

Multi-Sensor Camera(MAC address = 00:80:45:0d:00:01, channel number = 1),
Detection line={ Line1, 2}, Detection object = {Line1: Human, Line2: Vehicle},
Transmission interval = 5 min, Transmission time: {JST 2021/01/11 18:10:00}


```
{
   "CameraIPaddress":“192.168.0.10”,
   "CameraMACaddress":"00:80:45:0d:00:01",
   "Ch":"1",
   ”Time":”2021/1/11 9:10:00”,
   "TimeZone":"+0900",
   "SummerTime":0,
   "Line1":[
        {“list”: [
                [“2021/1/11 9:00”, 7, 6],
                [“2021/1/11 9:01”, 7, 7],
                [“2021/1/11 9:02”, 8, 10],
                [“2021/1/11 9:03”, 9, 11],
                [“2021/1/11 9:04”, 6, 5]
                   ]
      },
    ],
   “Line2”:[
        {“list”: [
                [“2021/1/11 9:00”, 4, 5],
                [“2021/1/11 9:01”, 5, 6],
                [“2021/1/11 9:02”, 12, 11],
                [“2021/1/11 9:03”, 12, 10],
                [“2021/1/11 9:04”, 9, 8]
        },
    ],
   “Line3”:[
        {“list”: []},
    ],

      ~

  “Line8”:[
        {“list”: []},
  ],
  “Line1_cntobj“:[“Human“],
  “Line2_cntobj“:[“Vehicle“],
  “Line3_cntobj“:[],

      ~

  “Line8_cntobj“:[]
}
```

Camera except for multi-sensor camera (MAC address = 00:80:45:0d:00:01),
Detection line={ Line1, 3}, Detection object = {Line1: Human, Line3: Bike},
Transmission interval = 1 min, Transmission time: {JST 2021/01/11 18:10:00}


```
{
   "CameraIPaddress”:“192.168.0.10”,
   "CameraMACaddress":"00:80:45:0d:00:01",
   ”Time":”2021/1/11 9:10:00”,
   "TimeZone":"+0900",
   "SummerTime":0,
   "Line1":[
         {“list”: [
                  [“2021/1/11 9:00”, 7, 6]
                ]
      },
    ],
   “Line2”:[
         {“list”: []},
    ],
   “Line3”:[
         {“list”: [
                  [“2021/1/11 9:00”, 4, 5]
                ]
         },
    ],
   “Line4”:[
         {“list”: []},
    ],

      ~

  “Line8”:[
         {“list”: []},
   ],
  “Line1_cntobj“:[“Human“],
  “Line2_cntobj“:[],
  “Line3_cntobj“:[“Bike“],
  “Line4_cntobj“:[],

      ~

  “Line8_cntobj“:[]
}
```

8. MQTT periodic transmission

The camera outputs data using the MQTT protocol as MQTT client.

8.1. Setting specifications

It is necessary to make the following settings in advance with this application.

Items	Description
Transmission destination	Select whether to send or not * In order to transmit data using the MQTT protocol, it is necessary to enable MQTT setting of camera
Topic	Name of topic
QoS	QoS level(0, 1, 2) Retain: Select to save the last transmitted messages on the MQTT server
Transmission interval	1min, 5min, 10min, 15min, 30min, 60min

8.2. Detail of telegraphic protocol

This application transmits the following telegram as PUBLISH message.

Bit	7	6	5	4	3	2	1	0
1	Message Type				DUP Flag	QoS Level		Retain
2	Remaining Length							

[Data format of fixed header]

[Header part]

Parameter	Length	Values and comments
Fixed header		
Message Type	4-bit unsigned value	Message Type(0~15)
DUP Flag	1-bit	Flag for redelivering 0: Not redeliver 1: redeliver
QoS Level	2-bit	Quality of Service levels 0: At most once 1: At least once 2: Exactly once

Retain	1-bit	Flag for retaining 0: Off 1: On
Remaining Length	8-bit	The number of bytes left in the current packet, including variable header and payload data
Variable header		
Msg Len	16bit (MSB, LSB)	Length of payload
Topic Length	16bit (MSB, LSB)	Length of topic name
Topic	UTF-encoded string	Topic name
Message Identifier	16-bit unsigned integer(MSB, LSB)	Message ID

[Payload]

Parameter	Value	Notation	Description
CameraIPaddress	(0~255)(0~255) (0~255)(0~255)	Decimal number	Camera IP address
CameraMACaddress	(00~ff)(00~ff) (00~ff)(00~ff) (00~ff)(00~ff)	Hex number	Camera MAC address
Ch	1, 2, 3, 4	Decimal number	Channnel number of Multi-Sensor Camera *This is applied when the camera is Multi-Sensor Camera.
Time	Date and time(UTC)		Date and time Format: yyyymmddhhmmss e.g.) August 29, 2013 12:35:01 Japan time 20130829123501
TimeZone	01200~11300		Time difference from UTC Minus is represented by 0 and plus is represented by 1. e.g.) Osaka, Sapporo, Tokyo (time

			difference of 9 hours) 10900
SummerTime	0, 1		Daylight saving time setting 0:non-daylight saving time 1:daylight saving time
Line1_In_Total Line2_In_Total Line3_In_Total Line4_In_Total Line5_In_Total Line6_In_Total Line7_In_Total Line8_In_Total	0~65,535		Number of objects crossed the line to Direction “In” for each detection line and for transmission interval. * If the detection line is not set, send it blank.
Line1_Out_Total Line2_Out_Total Line3_Out_Total Line4_Out_Total Line5_Out_Total Line6_Out_Total Line7_Out_Total Line8_Out_Total	0~65,535		Number of objects crossed the line to Direction “Out” for each detection line and for transmission interval. * If the detection line is not set, send it blank.
Line1_CountObjHuman Line1_CountObjVehicle Line1_CountObjBike Line2_CountObjHuman Line2_CountObjVehicle Line2_CountObjBike Line3_CountObjHuman Line3_CountObjVehicle Line3_CountObjBike Line4_CountObjHuman Line4_CountObjVehicle Line4_CountObjBike Line5_CountObjHuman Line5_CountObjVehicle Line5_CountObjBike	0, 1		Target object to be counted (Human, Vehicle, Bike) 0: Not count 1: Count * If the detection line is not set, send it blank. e.g.) If human is the target of counting, it will be as follows. "Line1_CountObjHuman":"1" "Line1_CountObjVehicle":"0" "Line1_CountObjBike":"0"

Line6_CountObjHuman Line6_CountObjVehicle Line6_CountObjBike Line7_CountObjHuman Line7_CountObjVehicle Line7_CountObjBike Line8_CountObjHuman Line8_CountObjVehicle Line8_CountObjBike			If bike is the target of counting, it will be as follows. "Line1_CountObjHuman":"0" "Line1_CountObjVehicle":"0" "Line1_CountObjBike":"1"

8.3. Transmission format

Multi-Sensor Camera(MAC address = 00:80:45:0d:00:01, channel number = 1), Detection line={ Line1, 2}, Detection object = {Line1: Human, Line2: Vehicle}, Transmission interval = 5 min, Transmission time: {JST 2021/01/11 18:10:00}


```
{
  "CameraIPaddress":"192168000010",
   "CameraMACaddress":"0080450d0001",
   "Ch":"1",
   "Time":"20210111091000",
   "TimeZone":"10900",
   "SummerTime":"0",
   "Line1_In_Total":"32",
   "Line1_Out_Total":"33",
   "Line2_In_Total":"71",
   "Line2_Out_Total":"67",
   "Line3_In_Total":"",
   "Line3_Out_Total":"",
   "Line4_In_Total":"",
   "Line4_Out_Total":"",
   "Line5_In_Total":"",
   "Line5_Out_Total":"",
   "Line6_In_Total":"",
   "Line6_Out_Total":"",
   "Line7_In_Total":"",
   "Line7_Out_Total":"",
   "Line8_In_Total":"",
   "Line8_Out_Total":"",
   "Line1_CountObjHuman":"1",
   "Line1_CountObjVehicle":"0",
   "Line1_CountObjBike":"0",
   "Line2_CountObjHuman":"0",
   "Line2_CountObjVehicle":"1",
   "Line2_CountObjBike":"0",
   "Line3_CountObjHuman":"",
   "Line3_CountObjVehicle":"",
   "Line3_CountObjBike":"",
   "Line4_CountObjHuman":"",
   "Line4_CountObjVehicle":"",
   "Line4_CountObjBike":"",
   "Line5_CountObjHuman":"",
   "Line5_CountObjVehicle":"",
   "Line5_CountObjBike":"",
   "Line6_CountObjHuman":"",
   "Line6_CountObjVehicle":"",
   "Line6_CountObjBike":"",
   "Line7_CountObjHuman":"",
   "Line7_CountObjVehicle":"",
   "Line7_CountObjBike":"",
   "Line8_CountObjHuman":"",
   "Line8_CountObjVehicle":"",
   "Line8_CountObjBike":""
}
```

9. Appendix

9.1. Specifications of CGI parameters

Item	CGI parameter (key)	Definition of the key	CGI parameter (numeric)	Definition of the value
Mode selection				
Mode selection	mode_switch	Select angle mode	0, 1, 2	0: Unselected 1: Normal angle mode 2: Perpendicular angle mode
AI motion detection setting				
Area setting	sel_condition	Detection program number	1, 2	1: Detection program 1 2: Detection program 2
Save as a preference in *detection setting1* if not specifying sel_condition. The beginning of the preference name in the parameter is “c1_” for detection	det_fig1	Type of polygonal shape; number of vertexes; coordinate of the vertexes (detection area1) (800x450 coordinate system) *A line is deemed as a figure with 2 vertexes.	0/1＋ (1～F)＋ (0～799 ＋0～799)× max. 16 vertexes	0/1：Geometry (polygonal/rectangular/line) 1～F：number of vertexes (0～799＋0～799)× max. 16 vertexes ：bit string of coordinates data (up to 16 vertexes) *Both setting ranges are 0 to 799 considering the corridor settings.

setting1 and “c2_” for detection setting2. ex) “c1_det_fig1” when the parameter is “det_fig1” and is for detection setting1, whereas “2_det_fig2” if the parameter is for detection setting2.	det_fig1_direction	Specify the direction for the direction detection (detection area1)	u/ur/r/br/b/bl/l/ul	u: Upper, ur: Upper right, r: Right, br: Lower right, b: Lower, bl: Lower left, l: Left, ul: Upper left The followings apply when numbers are used, 1: Up, 2: Upper right, 3: Right, 4: Lower right, 5: Lower, 6: Lower left, 7: Left, 8: Upper left
	det_fig1_line	Specify the direction for cross line. (detection area1)	1, 2, 3	1：A→B 2：A←B 3：A⇔B The followings apply when numbers are used (for detection setting1) c1_det_fig1_direction=9：A→B c1_det_fig1_direction=10：A←B c1_det_fig1_direction=11：A⇔B
	det_fig1_mode	Detection mode (detection area1)	1, 2, 4, 16	1：Intruder 2：Loitering 4：Direction 16：Cross line
	det_fig1_human	Target object: human (detection area1)	0/1	0：deactivated, 1：activated The followings apply when numbers are used。 0: FALSE, 1:TRUE
	det_fig1_car	Target object: vehicle (detection area1)	0/1	0：deactivated, 1：activated The followings apply when numbers are used

				0: FALSE, 1:TRUE
	det_fig1_bike	Target object: bicycle (detection area1)	0/1	0：deactivated, 1：activated The followings apply when numbers are used。 0: FALSE, 1:TRUE
	det_fig2	Type of polygonal shape; number of vertexes; coordinate of the vertexes (detection area2) (800x450 coordinate system) *A line is deemed as a figure with 2 vertexes.	0/1＋ (1～F)＋ (0～799＋0～799)× max. 16 vertexes	0/1：Geometry (polygonal/rectangular/line) 1～F：number of vertexes (0～799＋0～799)× max. 16 vertexes ：bit string of coordinates data (up to 16 vertexes) *Both setting ranges are 0 to 799 considering the corridor settings.
	det_fig2_direction	Specify direction for the direction detection (detection area2)	u/ur/r/br/b/bl/l/ul	u: Upper, ur: Upper right, r: Right, br: Lower right, b: Lower, bl: Lower left, l: Left, ul: Upper left The followings apply when numbers are used, 1: Up, 2: Upper right, 3: Right, 4: Lower right, 5: Lower, 6: Lower left, 7: Left, 8: Upper left
	det_fig2_line	Direction setting for cross line (detection area2)	1, 2, 3	1：A→B 2：A←B 3：A⇔B

				The followings apply when numbers are used。 (for detection setting1) c1_det_fig2_direction=9：A→B c1_det_fig2_direction=10：A←B c1_det_fig2_direction=11：A⇔B
	det_fig2_mode	Detection mode (detection area2)	1, 2, 4, 16	1：Intruder 2：Loitering 4：Direction 16：Cross line
	det_fig2_human	Target object: human (detection area2)	0/1	0：deactivated, 1：activated The followings apply when numbers are used。 0: FALSE, 1:TRUE
	det_fig2_car	Target object: vehicle (detection area2)	0/1	0：deactivated, 1：activated The followings apply when numbers are used。 0: FALSE, 1:TRUE
	det_fig2_bike	Target object: bicycle (detection area2)	0/1	0：deactivated, 1：activated The followings apply when numbers are used。 0: FALSE, 1:TRUE
	det_fig3	Type of polygonal shape; number of vertexes; coordinate of the vertexes (detection area3)	0/1＋ (1～F)＋ (0～799＋0～799)× max. 16 vertexes	0/1：Geometry (polygonal/rectangular/line) 1～F：number of vertexes (0～799＋0～799)× max. 16 vertexes ：bit string of coordinates data (up to 16 vertexes)

		(800x450 coordinate system) *A line is deemed as a figure with 2 vertexes.		*Both setting ranges are 0 to 799 considering the corridor settings.
	det_fig3_direction	Specify direction for the direction detection (detection area3)	u/ur/r/br/b/bl/l/ul	u: Upper, ur: Upper right, r: Right, br: Lower right, b: Lower, bl: Lower left, l: Left, ul: Upper left The followings apply when numbers are used, 1: Up, 2: Upper right, 3: Right, 4: Lower right, 5: Lower, 6: Lower left, 7: Left, 8: Upper left
	det_fig3_line	Direction setting for cross line (detection area3)	1, 2, 3	1：A→B 2：A←B 3：A⇔B The followings apply when numbers are used。 (for detection setting1) c1_det_fig3_direction=9：A→B c1_det_fig3_direction=10：A←B c1_det_fig3_direction=11：A⇔B
	det_fig3_mode	Detection mode (detection area3)	1, 2, 4, 16	1：Intruder 2：Loitering 4：Direction 16：Cross line

	det_fig3_human	Target object: human (detection area3)	0/1	0：deactivated, 1：activated The followings apply when numbers are used。 0: FALSE, 1:TRUE
	det_fig3_car	Target object: vehicle (detection area3)	0/1	0：deactivated, 1：activated The followings apply when numbers are used。 0: FALSE, 1:TRUE
	det_fig3_bike	Target object: bicycle (detection area3)	0/1	0：deactivated, 1：activated The followings apply when numbers are used。 0: FALSE, 1:TRUE
	det_fig4	Type of polygonal shape; number of vertexes; coordinate of the vertexes (detection area4) (800x450 coordinate system) *A line is deemed as a figure with 2 vertexes.	0/1＋ (1～F)＋ (0～799 ＋0～799)× max. 16 vertexes	0/1：Geometry (polygonal/rectangular/line) 1～F：number of vertexes (0～799＋0～799)× max. 16 vertexes ：bit string of coordinates data (up to 16 vertexes) *Both setting ranges are 0 to 799 considering the corridor settings.
	det_fig4_direction	Specify direction for the direction detection (detection area4)	u/ur/r/br/b/bl/l/ul	u: Upper, ur: Upper right, r: Right, br: Lower right, b: Lower, bl: Lower left, l: Left, ul: Upper left The followings apply when numbers are used, 1: Up, 2: Upper right, 3: Right, 4: Lower right, 5: Lower, 6: Lower left, 7: Left, 8: Upper left

	det_fig4_line	Direction setting for cross line (detection area4)	1, 2, 3	1：A→B 2：A←B 3：A⇔B The followings apply when numbers are used。 (for detection setting1) c1_det_fig4_direction=9：A→B c1_det_fig4_direction=10：A←B c1_det_fig4_direction=11：A⇔B
	det_fig4_mode	Detection mode (detection area4)	1, 2, 4, 16	1：Intruder 2：Loitering 4：Direction 16：Cross line
	det_fig4_human	Target object: human (detection area4)	0/1	0：deactivated, 1：activated The followings apply when numbers are used。 0: FALSE, 1:TRUE
	det_fig4_car	Target object: vehicle (detection area4)	0/1	0：deactivated, 1：activated The followings apply when numbers are used。 0: FALSE, 1:TRUE
	det_fig4_bike	Target object: bicycle (detection area4)	0/1	0：deactivated, 1：activated The followings apply when numbers are used。 0: FALSE, 1:TRUE

	det_fig5	Type of polygonal shape; number of vertexes; coordinate of the vertexes (detection area5) (800x450 coordinate system) *A line is deemed as a figure with 2 vertexes.	0/1＋ (1～F)＋ (0～799 ＋0～799)× max. 16 vertexes	0/1：Geometry (polygonal/rectangular/line) 1～F：number of vertexes (0～799＋0～799)× max. 16 vertexes ：bit string of coordinates data (up to 16 vertexes) *Both setting ranges are 0 to 799 considering the corridor settings.
	det_fig5_direction	Specify direction for the direction detection (detection area5)	u/ur/r/br/b/bl/l/ul	u: Upper, ur: Upper right, r: Right, br: Lower right, b: Lower, bl: Lower left, l: Left, ul: Upper left The followings apply when numbers are used, 1: Up, 2: Upper right, 3: Right, 4: Lower right, 5: Lower, 6: Lower left, 7: Left, 8: Upper left
	det_fig5_line	Direction setting for cross line (detection area5)	1, 2, 3	1：A→B 2：A←B 3：A⇔B The followings apply when numbers are used。 (for detection setting1) c1_det_fig5_direction=9：A→B c1_det_fig5_direction=10：A←B c1_det_fig5_direction=11：A⇔B

	det_fig5_mode	Detection mode (detection area5)	1, 2, 4, 16	1：Intruder 2：Loitering 4：Direction 16：Cross line
	det_fig5_human	Target object: human (detection area5)	0/1	0：deactivated, 1：activated The followings apply when numbers are used。 0: FALSE, 1:TRUE
	det_fig5_car	Target object: vehicle (detection area5)	0/1	0：deactivated, 1：activated
	det_fig5_bike	Target object: bicycle (detection area5)	0/1	0：deactivated, 1：activated The followings apply when numbers are used。 0: FALSE, 1:TRUE
	det_fig6	Type of polygonal shape; number of vertexes; coordinate of the vertexes (detection area6) (800x450 coordinate system) *A line is deemed as a figure with 2 vertexes.	0/1＋ (1～F)＋ (0～799 ＋0～799)× max. 16 vertexes	0/1：Geometry (polygonal/rectangular/line) 1～F：number of vertexes (0～799＋0～799)× max. 16 vertexes ：bit string of coordinates data (up to 16 vertexes) *Both setting ranges are 0 to 799 considering the corridor settings.

	det_fig6_direction	Specify direction for the direction detection (detection area6)	u/ur/r/br/b/bl/l/ul	u: Upper, ur: Upper right, r: Right, br: Lower right, b: Lower, bl: Lower left, l: Left, ul: Upper left The followings apply when numbers are used, 1: Up, 2: Upper right, 3: Right, 4: Lower right, 5: Lower, 6: Lower left, 7: Left, 8: Upper left
	det_fig6_line	Direction setting for cross line (detection area6)	1, 2, 3	1：A→B 2：A←B 3：A⇔B The followings apply when numbers are used。 (for detection setting1) c1_det_fig6_direction=9：A→B c1_det_fig6_direction=10：A←B c1_det_fig6_direction=11：A⇔B
	det_fig6_mode	Detection mode (detection area6)	1, 2, 4, 16	1：Intruder 2：Loitering 4：Direction 16：Cross line
	det_fig6_human	Target object: human (detection area6)	0/1	0：deactivated, 1：activated The followings apply when numbers are used。 0: FALSE, 1:TRUE
	det_fig6_car	Target object: vehicle (detection area6)	0/1	0：deactivated, 1：activated The followings apply when numbers are used。 0: FALSE, 1:TRUE

	det_fig6_bike	Target object: bicycle (detection area6)	0/1	0：deactivated, 1：activated The followings apply when numbers are used。 0: FALSE, 1:TRUE
	det_fig7	Type of polygonal shape; number of vertexes; coordinate of the vertexes (detection area7) (800x450 coordinate system) *A line is deemed as a figure with 2 vertexes.	0/1＋ (1～F)＋ (0～799 ＋0～799)× max. 16 vertexes	0/1：Geometry (polygonal/rectangular/line) 1～F：number of vertexes (0～799＋0～799)× max. 16 vertexes ：bit string of coordinates data (up to 16 vertexes) *Both setting ranges are 0 to 799 considering the corridor settings.
	det_fig7_direction	Specify direction for the direction detection (detection area7)	u/ur/r/br/b/bl/l/ul	u: Upper, ur: Upper right, r: Right, br: Lower right, b: Lower, bl: Lower left, l: Left, ul: Upper left The followings apply when numbers are used, 1: Up, 2: Upper right, 3: Right, 4: Lower right, 5: Lower, 6: Lower left, 7: Left, 8: Upper left
	det_fig7_line	Direction setting for cross line (detection area7)	1, 2, 3	1：A→B 2：A←B 3：A⇔B The followings apply when numbers are used。 (for detection setting1)

				c1_det_fig7_direction=9：A→B c1_det_fig7_direction=10：A←B c1_det_fig7_direction=11：A⇔B
	det_fig7_mode	Detection mode (detection area7)	1, 2, 4, 16	1：Intruder 2：Loitering 4：Direction 16：Cross line
	det_fig7_human	Target object: human (detection area7)	0/1	0：deactivated, 1：activated The followings apply when numbers are used。 0: FALSE, 1:TRUE
	det_fig7_car	Target object: vehicle (detection area7)	0/1	0：deactivated, 1：activated The followings apply when numbers are used。 0: FALSE, 1:TRUE
	det_fig7_bike	Target object: bicycle (detection area7)	0/1	0：deactivated, 1：activated The followings apply when numbers are used。 0: FALSE, 1:TRUE
	det_fig8	Type of polygonal shape; number of vertexes; coordinate of the vertexes (detection area8) (800x450 coordinate system)	0/1＋ (1～F)＋ (0～799 ＋0～799)× max. 16 vertexes	0/1：Geometry (polygonal/rectangular/line) 1～F：number of vertexes (0～799＋0～799)× max. 16 vertexes ：bit string of coordinates data (up to 16 vertexes) *Both setting ranges are 0 to 799 considering the corridor settings.

		*A line is deemed as a figure with 2 vertexes.		
	det_fig8_direction	Specify direction for the direction detection (detection area8)	u/ur/r/br/b/bl/l/ul	u: Upper, ur: Upper right, r: Right, br: Lower right, b: Lower, bl: Lower left, l: Left, ul: Upper left The followings apply when numbers are used, 1: Up, 2: Upper right, 3: Right, 4: Lower right, 5: Lower, 6: Lower left, 7: Left, 8: Upper left
	det_fig8_line	Direction setting for cross line (detection area8)	1, 2, 3	1：A→B 2：A←B 3：A⇔B The followings apply when numbers are used。 (for detection setting1) c1_det_fig8_direction=9：A→B c1_det_fig8_direction=10：A←B c1_det_fig8_direction=11：A⇔B
	det_fig8_mode	Detection mode (detection area8)	1, 2, 4, 16	1：Intruder 2：Loitering 4：Direction 16：Cross line
	det_fig8_human	Target object: human (detection area8)	0/1	0：deactivated, 1：activated The followings apply when numbers are used。

				0: FALSE, 1:TRUE
	det_fig8_car	Target object: vehicle (detection area8)	0/1	0：deactivated, 1：activated The followings apply when numbers are used。 0: FALSE, 1:TRUE
	det_fig8_bike	Target object: bicycle (detection area8)	0/1	0：deactivated, 1：activated The followings apply when numbers are used。 0: FALSE, 1:TRUE
	mask_fig1	Type of polygonal shape; number of vertexes; coordinate of the vertexes for the 1st mask area (800x450 coordinate system)	0/1＋ (1～F)＋ (0～799 ＋0～799)× max. 16 vertexes	0/1：Geometry (polygonal/rectangular/line) 1～F：number of vertexes (0～799＋0～799)× max. 16 vertexes ：bit string of coordinates data (up to 16 vertexes) *Both setting ranges are 0 to 799 considering the corridor settings.
	mask_fig2	Type of polygonal shape; number of vertexes; coordinate of the vertexes for the 2nd mask area (800x450 coordinate system)	0/1＋ (1～F)＋ (0～799 ＋0～449)× max. 16 vertexes	0/1：Geometry (polygonal/rectangular/line) 1～F：number of vertexes (0～799＋0～799)× max. 16 vertexes ：bit string of coordinates data (up to 16 vertexes) *Both setting ranges are 0 to 799 considering the corridor settings.

	mask_fig3	Type of polygonal shape; number of vertexes; coordinate of the vertexes for the 3rd mask area (800x450 coordinate system)	0/1＋ (1～F)＋ (0～799 ＋0～449)× max. 16 vertexes	0/1：Geometry (polygonal/rectangular/line) 1～F：number of vertexes (0～799＋0～799)× max. 16 vertexes ：bit string of coordinates data (up to 16 vertexes) *Both setting ranges are 0 to 799 considering the corridor settings.
	mask_fig4	Type of polygonal shape; number of vertexes; coordinate of the vertexes for the 4th mask area (800x450 coordinate system)	0/1＋ (1～F)＋ (0～799 ＋0～449)× max. 16 vertexes	0/1：Geometry (polygonal/rectangular/line) 1～F：number of vertexes (0～799＋0～799)× max. 16 vertexes ：bit string of coordinates data (up to 16 vertexes) *Both setting ranges are 0 to 799 considering the corridor settings.
	mask_fig5	Type of polygonal shape; number of vertexes; coordinate of the vertexes for the 5th mask area (800x450 coordinate system)	0/1＋ (1～F)＋ (0～799 ＋0～449)× max. 16 vertexes	0/1：Geometry (polygonal/rectangular/line) 1～F：number of vertexes (0～799＋0～799)× max. 16 vertexes ：bit string of coordinates data (up to 16 vertexes) *Both setting ranges are 0 to 799 considering the corridor settings.

	mask_fig6	Type of polygonal shape; number of vertexes; coordinate of the vertexes for the 6th mask area (800x450 coordinate system)	0/1＋ (1～F)＋ (0～799 ＋0～449)× max. 16 vertexes	0/1：Geometry (polygonal/rectangular/line) 1～F：number of vertexes (0～799＋0～799)× max. 16 vertexes ：bit string of coordinates data (up to 16 vertexes) *Both setting ranges are 0 to 799 considering the corridor settings.
	mask_fig7	Type of polygonal shape; number of vertexes; coordinate of the vertexes for the 7th mask area (800x450 coordinate system)	0/1＋ (1～F)＋ (0～799 ＋0～449)× max. 16 vertexes	0/1：Geometry (polygonal/rectangular/line) 1～F：number of vertexes (0～799＋0～799)× max. 16 vertexes ：bit string of coordinates data (up to 16 vertexes) *Both setting ranges are 0 to 799 considering the corridor settings.
	mask_fig8	Type of polygonal shape; number of vertexes; coordinate of the vertexes for the 8th mask area (800x450 coordinate system)	0/1＋ (1～F)＋ (0～799 ＋0～449)× max. 16 vertexes	0/1：Geometry (polygonal/rectangular/line) 1～F：number of vertexes (0～799＋0～799)× max. 16 vertexes ：bit string of coordinates data (up to 16 vertexes) *Both setting ranges are 0 to 799 considering the corridor settings.

Count function setting				
Count function On/Off	info_switch	Count function On/Off	0/1	0: Off 1: On
Line settings	det_fig1	Coordinate of the vertexes (detection line1)	21XXXXYYYYXXXXYYYY (XXXX,YYYY)-(XXXX,YYYY) is the coordinate of detection line	Representing the two coordinates of detection line in decimal. E.g.) The coordinates are (120,130)-(240,222), the value is 210120013002400222
	det_fig2	Coordinate of the vertexes (detection line2)	Same as above	Same as above
	det_fig3	Coordinate of the vertexes (detection line3)	Same as above	Same as above
	det_fig4	Coordinate of the vertexes (detection line4)	Same as above	Same as above
	det_fig5	Coordinate of the vertexes (detection line5)	Same as above	Same as above

	det_fig6	Coordinate of the vertexes (detection line6)	Same as above	Same as above
	det_fig7	Coordinate of the vertexes (detection line7)	Same as above	Same as above
	det_fig8	Coordinate of the vertexes (detection line8)	Same as above	Same as above
	det_fig1_human	Target object: human (detection line1)	0/1	0：deactivated, 1：activated The followings apply when numbers are used。 0: FALSE, 1:TRUE
	det_fig1_car	Target object: vehicle (detection line1)	0/1	0：deactivated, 1：activated The followings apply when numbers are used 0: FALSE, 1:TRUE
	det_fig1_bike	Target object: bicycle (detection line1)	0/1	0：deactivated, 1：activated The followings apply when numbers are used。 0: FALSE, 1:TRUE
	det_fig2_human	Target object: human (detection line2)	0/1	0：deactivated, 1：activated The followings apply when numbers are used。 0: FALSE, 1:TRUE
	det_fig2_car	Target object: vehicle (detection line2)	0/1	0：deactivated, 1：activated The followings apply when numbers are used

				0: FALSE, 1:TRUE
	det_fig2_bike	Target object: bicycle (detection line2)	0/1	0：deactivated, 1：activated The followings apply when numbers are used。 0: FALSE, 1:TRUE
	det_fig3_human	Target object: human (detection line3)	0/1	0：deactivated, 1：activated The followings apply when numbers are used。 0: FALSE, 1:TRUE
	det_fig3_car	Target object: vehicle (detection line3)	0/1	0：deactivated, 1：activated The followings apply when numbers are used 0: FALSE, 1:TRUE
	det_fig3_bike	Target object: bicycle (detection line3)	0/1	0：deactivated, 1：activated The followings apply when numbers are used。 0: FALSE, 1:TRUE
	det_fig4_human	Target object: human (detection line4)	0/1	0：deactivated, 1：activated The followings apply when numbers are used。 0: FALSE, 1:TRUE
	det_fig4_car	Target object: vehicle (detection line4)	0/1	0：deactivated, 1：activated The followings apply when numbers are used 0: FALSE, 1:TRUE
	det_fig4_bike	Target object: bicycle (detection line4)	0/1	0：deactivated, 1：activated The followings apply when numbers are used。 0: FALSE, 1:TRUE

	det_fig5_human	Target object: human (detection line5)	0/1	0：deactivated, 1：activated The followings apply when numbers are used。 0: FALSE, 1:TRUE
	det_fig5_car	Target object: vehicle (detection line5)	0/1	0：deactivated, 1：activated The followings apply when numbers are used 0: FALSE, 1:TRUE
	det_fig5_bike	Target object: bicycle (detection line5)	0/1	0：deactivated, 1：activated The followings apply when numbers are used。 0: FALSE, 1:TRUE
	det_fig6_human	Target object: human (detection line6)	0/1	0：deactivated, 1：activated The followings apply when numbers are used。 0: FALSE, 1:TRUE
	det_fig6_car	Target object: vehicle (detection line6)	0/1	0：deactivated, 1：activated The followings apply when numbers are used 0: FALSE, 1:TRUE
	det_fig6_bike	Target object: bicycle (detection line6)	0/1	0：deactivated, 1：activated The followings apply when numbers are used。 0: FALSE, 1:TRUE
	det_fig7_human	Target object: human (detection line7)	0/1	0：deactivated, 1：activated The followings apply when numbers are used。 0: FALSE, 1:TRUE
	det_fig7_car	Target object: vehicle (detection line7)	0/1	0：deactivated, 1：activated The followings apply when numbers are used

				0: FALSE, 1:TRUE
	det_fig7_bike	Target object: bicycle (detection line7)	0/1	0：deactivated, 1：activated The followings apply when numbers are used。 0: FALSE, 1:TRUE
	det_fig8_human	Target object: human (detection line8)	0/1	0：deactivated, 1：activated The followings apply when numbers are used。 0: FALSE, 1:TRUE
	det_fig8_car	Target object: vehicle (detection line8)	0/1	0：deactivated, 1：activated The followings apply when numbers are used 0: FALSE, 1:TRUE
	det_fig8_bike	Target object: bicycle (detection line8)	0/1	0：deactivated, 1：activated The followings apply when numbers are used。 0: FALSE, 1:TRUE
	det_fig1_stat	Activated/Deactivated (detection line1)	0/1	0：deactivated, 1：activated
	det_fig2_stat	Activated/Deactivated (detection line2)	0/1	0：deactivated, 1：activated
	det_fig3_stat	Activated/Deactivated (detection line3)	0/1	0：deactivated, 1：activated
	det_fig4_stat	Activated/Deactivated (detection line4)	0/1	0：deactivated, 1：activated
	det_fig5_stat	Activated/Deactivated (detection line5)	0/1	0：deactivated, 1：activated

	det_fig6_stat	Activated/Deactivated (detection line6)	0/1	0：deactivated, 1：activated
	det_fig7_stat	Activated/Deactivated (detection line7)	0/1	0：deactivated, 1：activated
	det_fig8_stat	Activated/Deactivated (detection line8)	0/1	0：deactivated, 1：activated
	det_fig1_line	Specify direction for the cross line detection (detection line1)	1,2,3	1：In 2：Out 3：In/Out
	det_fig2_line	Specify direction for the cross line detection (detection line2)	1,2,3	1：In 2：Out 3：In/Out
	det_fig3_line	Specify direction for the cross line detection (detection line3)	1,2,3	1：In 2：Out 3：In/Out
	det_fig4_line	Specify direction for the cross line detection (detection line4)	1,2,3	1：In 2：Out 3：In/Out
	det_fig5_line	Specify direction for the cross line detection (detection line5)	1,2,3	1：In 2：Out 3：In/Out

	det_fig6_line	Specify direction for the cross line detection (detection line6)	1,2,3	1：In 2：Out 3：In/Out
	det_fig7_line	Specify direction for the cross line detection (detection line7)	1,2,3	1：In 2：Out 3：In/Out
	det_fig8_line	Specify direction for the cross line detection (detection line8)	1,2,3	1：In 2：Out 3：In/Out
HTTP transmission	notify1	Transmission ON/OFF (Transmission destinasion 1)	0,1	0：Off 1：On
	addr1	Transmission address (Transmission destinasion 1)	(0~255)+.+(0~255)+.+(0~255)+.+(0~255) Half-width alphanumeric, half-width symbol[._-] and alphabet within 63 characters	(0~255)+.+(0~255)+.+(0~255)+.+(0~255): IP address Half-width alphanumeric: host name
	path1	Transmission path name (Transmission destinasion 1)	Half-width alphanumeric and half-width symbol within 128 characters	Path name of the transmission destination E.g. /AIVMDApp

	ssl1	SSL (Transmission destinasion 1)	0,1	0: Off 1: On
	port1	Transmission port number (Transmission destinasion 1)	1~65535	
	usr1	User name (Transmission destinasion 1)	Half-width alphanumeric within 128 characters	
	pass1	Password (Transmission destinasion 1)	Half-width alphanumeric within 63 characters	
	interval1	Transmission interval(min) (Transmission destinasion 1)	1,5,10,15,30,60	1, 5, 10, 15, 30, 60: Transmission interval of metadata (min)
	notify2	Transmission ON/OFF (Transmission destinasion 1)	0,1	0：Off 1：On
	addr2	Transmission address (Transmission destinasion 1)	(0~255)+.+(0~255)+.+(0~255)+.+(0~255) Half-width alphanumeric,	(0~255)+.+(0~255)+.+(0~255)+.+(0~255): IP address Half-width alphanumeric: host name

			half-width symbol[._-] and alphabet within 63 characters	
	path2	Transmission path name (Transmission destinasion 1)	Half-width alphanumeric and half-width symbol within 128 characters	Path name of the transmission destination E.g. /AIVMDApp
	ssl2	SSL (Transmission destinasion 1)	0,1	0: Off 1: On
	port2	Transmission port number (Transmission destinasion 1)	1~65535	
	usr2	User name (Transmission destinasion 1)	Half-width alphanumeric within 128 characters	
	pass2	Password (Transmission destinasion 1)	Half-width alphanumeric within 63 characters	
	interval2	Transmission interval(min) (Transmission destinasion 1)	1,5,10,15,30,60	1, 5, 10, 15, 30, 60: Transmission interval of metadata (min)

	notify3	Transmission ON/OFF (Transmission destinasion 1)	0,1	0：Off 1：On
	addr3	Transmission address (Transmission destinasion 1)	(0~255)+.+(0~255)+.+(0~255)+.+(0~255) Half-width alphanumeric, half-width symbol[._-] and alphabet within 63 characters	(0~255)+.+(0~255)+.+(0~255)+.+(0~255): IP address Half-width alphanumeric: host name
	path3	Transmission path name (Transmission destinasion 1)	Half-width alphanumeric and half-width symbol within 128 characters	Path name of the transmission destination E.g. /AIVMDApp
	ssl3	SSL (Transmission destinasion 1)	0,1	0: Off 1: On
	port3	Transmission port number (Transmission destinasion 1)	1~65535	
	usr3	User name (Transmission destinasion 1)	Half-width alphanumeric within 128 characters	

	pass3	Password (Transmission destinasion 1)	Half-width alphanumeric within 63 characters	
	interval3	Transmission interval(min) (Transmission destinasion 1)	1,5,10,15,30,60	1, 5, 10, 15, 30, 60: Transmission interval of metadata (min)
	notify4	Transmission ON/OFF (Transmission destinasion 1)	0,1	0：Off 1：On
	addr4	Transmission address (Transmission destinasion 1)	(0~255)+.+(0~255)+.+(0~255)+.+(0~255) Half-width alphanumeric, half-width symbol[._-] and alphabet within 63 characters	(0~255)+.+(0~255)+.+(0~255)+.+(0~255): IP address Half-width alphanumeric: host name
	path4	Transmission path name (Transmission destinasion 1)	Half-width alphanumeric and half-width symbol within 128 characters	Path name of the transmission destination E.g. /AIVMDApp
	ssl4	SSL (Transmission destinasion 1)	0,1	0: Off 1: On

	port4	Transmission port number (Transmission destinasion 1)	1~65535	
	usr4	User name (Transmission destinasion 1)	Half-width alphanumeric within 128 characters	
	pass4	Password (Transmission destinasion 1)	Half-width alphanumeric within 63 characters	
	interval4	Transmission interval(min) (Transmission destinasion 1)	1,5,10,15,30,60	1, 5, 10, 15, 30, 60: Transmission interval of metadata (min)
MQTT transmission	mqtt_notify	Transmission ON/OFF	0,1	0：Off 1：On
	mqtt_topic	Topic	Half-width alphanumeric and half-width symbol within 128 characters	E.g. i-PRO/NetworkCamera/App/AIVMD
	mqtt_qos	QoS	0, 1, 2	0: At most once 1: At least once 2: Exactly once

	mqtt_retain	Retain flag	0,1	0：Off 1：On
	mqtt_interval	Transmission interval	1,5,10,15,30,60	1, 5, 10, 15, 30, 60：Transmission interval of meta information (min)
Other	rec_interval	Count data storage interval	15,60,720,1440	15: 15 min 60: 1 hour 720: 12 hours 1440: 24 hours
	info_interval	The interval for the transmission of ONVIF®	0,1	0: 15 secounds 1: 1 min
Detailed setteing				
AI-VMD alarm	aivmd_alm_frequency	Alert frequency of AI-VMD alarm	0, 1	0: An alarm is issued only when a motion is detected for the first time 1: An alarm is issued every time a motion is detected
Sensitivity setting	sens_level	Motion detection sensitivity	1～7	1～7
	human_level	Human identification sensitivity	1～99	
	car_level	Vehicle identification sensitivity	1～99	
	bike_level	Bicycle identification sensitivity	1～99	

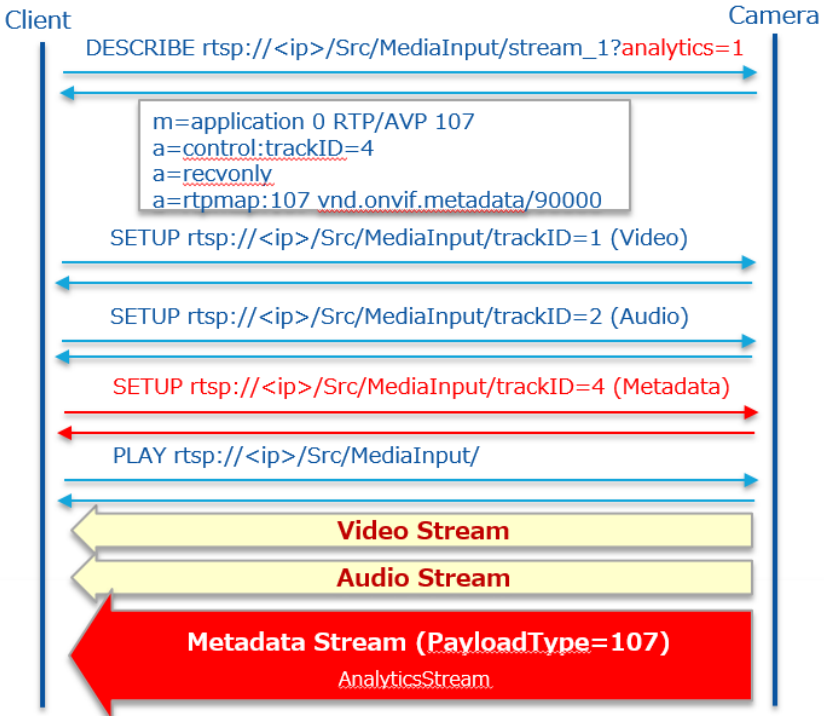
Time setting	intruder_time	Intruder detection time	0.2/0.4/1/2/3/4/5/10	0.2s／0.4s／1s／2s／3s／4s／5s／10s The followings apply when numbers are used。 2 (0.2s)/4 (0.4s)/10 (1s)/20 (2s)/30 (3s)/40 (4s)/50 (5s)/100 (10s)
	loitering_time	Loitering detection time	10/20/30/60/120	10s／20s／30s／1min／2min
	direction_time	Direction detection time	1/2/3/4/5/10	1s／2s／3s／4s／5s／10s
Depth setting	use_inch_as_unit	Use “inche” as unit	1, 0	Use inches (ticked): 1 Use cm (unticked): 0
	ivmd3d_object1_size	Photographic subject size (marker1)	100-200	
	ivmd3d_object2_size	Photographic subject size (marker2)	100-200	
	ivmd3d_size_max	Max. size	100～300/ disable	100%／150%／200%／250%／300%／ disable: unlimited
	ivmd3d_size_max	Min. size	100～400/ disable	100%／150%／200%／250%／300%／350／400%／ disable: unlimited
AI-VMD information addition	ivmd_info	AI-VMD information addition	0, 1, 2, 3	0: Off 1: On 2: On(with live display [with blue frame]) 3: On (with live display)
Additional information type	ivmd_info_type	Additional information type	0, 1, 2	0: Without detected object information 1: With detected object information (alarm frame

				information) 2: With detected object information (AI frame information)
Initialize settings	initialize	Setting data initialization	1	1: Initialize If not 1: Do not initialize

9.2. ONVIF Meta Stream transmission sequence

【RTSP URL】

・Send “analytics=1”, “event=1”, or “analytics=1&event=1” respectively when requesting (RTSP URL) Analytics Stream, Event Stream, or both.

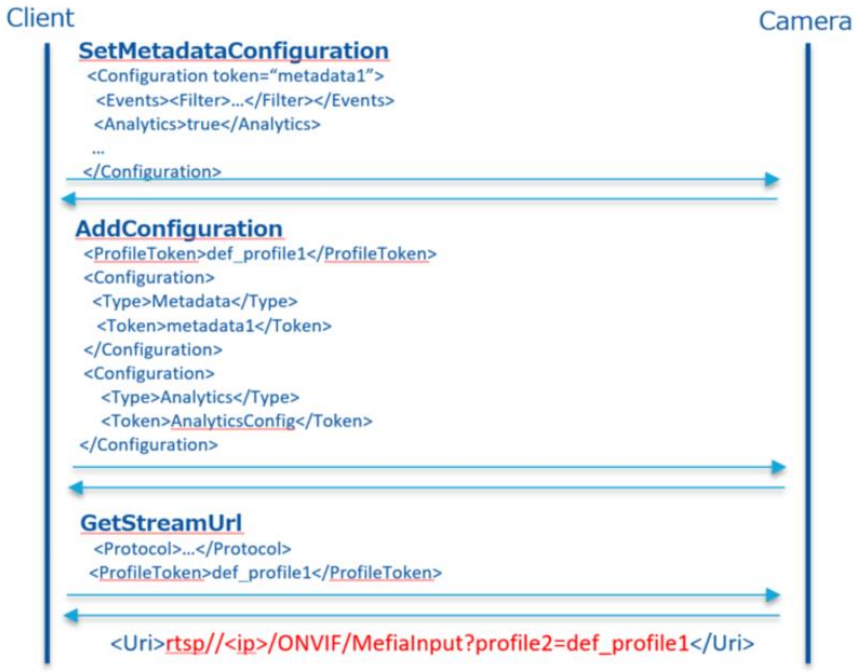


【ONVIF】

* 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