go1984 Performance Optimization

Date: March 2019
Based on go1984 version 8.0.0.1

go1984 Performance Optimization

http://www.go1984.com

Alfred-Mozer-Str. 51
D-48527 Nordhorn
Germany

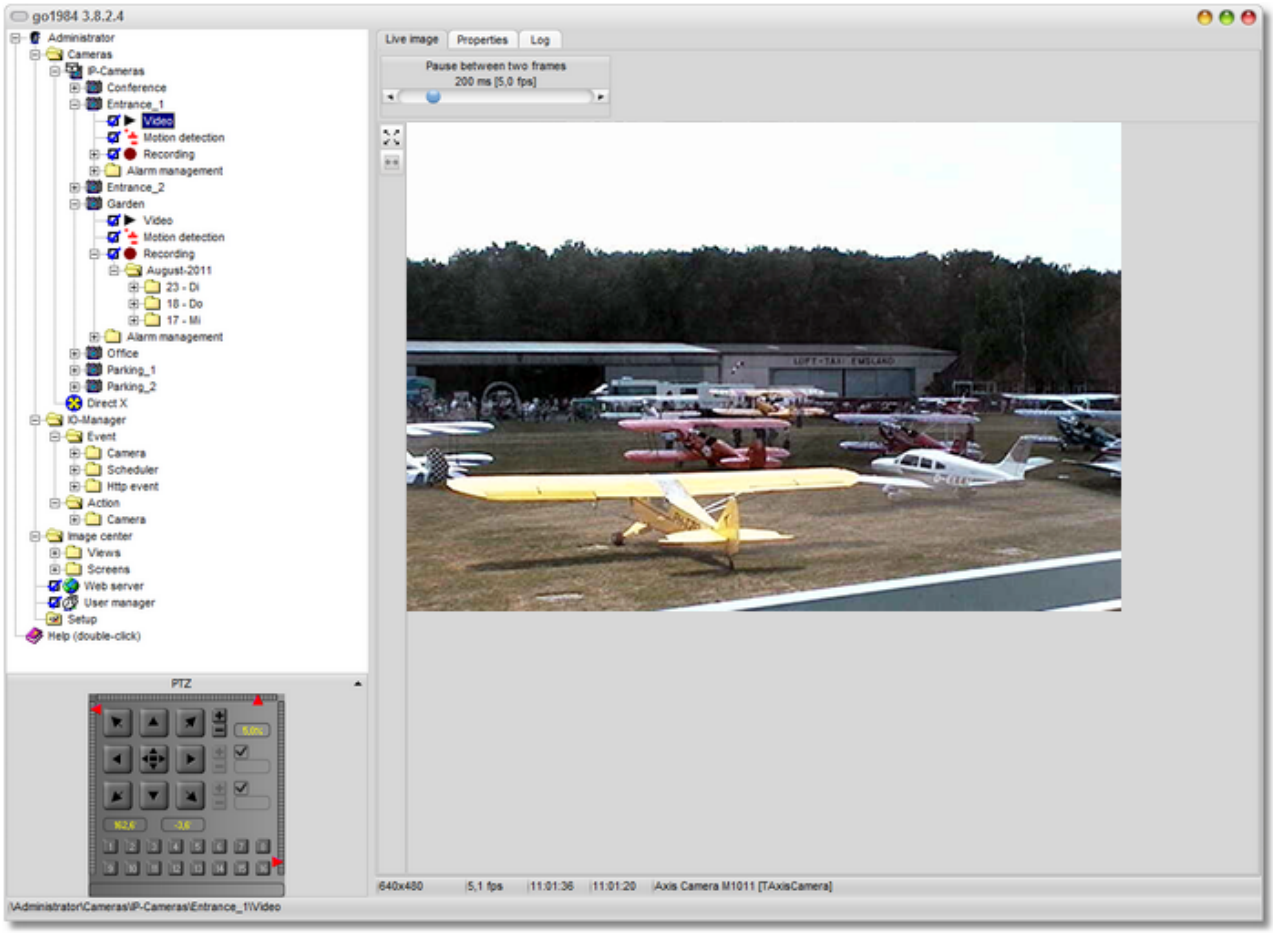
Telephone: +49 (0)5921 7139925
Fax: +49 (0)5921 7139929

http://www.logiware.de

Table of Contents

1 Introduction

go1984 offers high performance and extensive functionality for problem-free, professional video surveillance.



During the development of go1984 the main focus was on the performance of the system.

The default settings in go1984 have been chosen to ensure that you can use the system intuitively without extensive knowledge, that the system is stable and reliable and that you can add cameras easily.

Our experience shows that in 95% of all go1984 installations decompression and compression of JPG images is responsible for more than 90% of the CPU usage.

Against this background, the purpose of this document is to describe the methods you can use to improve the performance of your system further. It explains the influence of the image size and pause between two frames, the fundamentals of graphics formats and the benefits and disadvantages of different image sources.

Specific examples using MJPEG sources and megapixel cameras provide valuable information on simple methods of improving the overall performance of your go1984 surveillance system.

Bear in mind, in any case, that you should choose the go1984 version to suit your purposes as the Ultimate Edition has a performance level that is twice the level of the Enterprise Edition and four times that of the Pro Edition. You can also obtain further information about this by clicking the following link: https://www.go1984.com/software-overview/edition-comparison.html.

Feature	Pro	Enterprise	Ultimate
Recommended max. number of cameras (1)	16	32	64
Performance indicator	1	2	4
max. FPS at 720p/1080p (MJpeg)	130/60	260/120	520/240
recommended CPU e.g.	i3-7100/8100	i5-7400/8400	i7-7700/8700
max. camera resolution MJpeg streaming	unlimited	unlimited	unlimited
Processing of H.264 up to max. resolution (2)	1920x1080	2688x1520	3840x2160
Simultaneous client connections	2	unlimited	unlimited
Client https connections ("Let's Encrypt" SSL certificates)	✓	✓	✓
Support for third-party SSL certificates	-	-	✓
Can run as service	-	✓	✓
Time based ring storage	-	✓	✓
Program start in case of alarm	-	✓	✓
PTZ guard-tour support	-	✓	✓
Max. number of devices for push notification (3)	2	8	64

(1) Due to system restrictions max. 16/32/64 megapixel overall (MJpeg) recommended in Pro/Enterprise/Ultimate edition.

(2) H.264 is available in all editions up to 4K (UHD). Further processing is carried out depending on the edition up to the above mentioned max. resolution.

(3) Number of push notifications is limited to 100 messages per device per day. Precondition for push notification is an active go1984 update service.

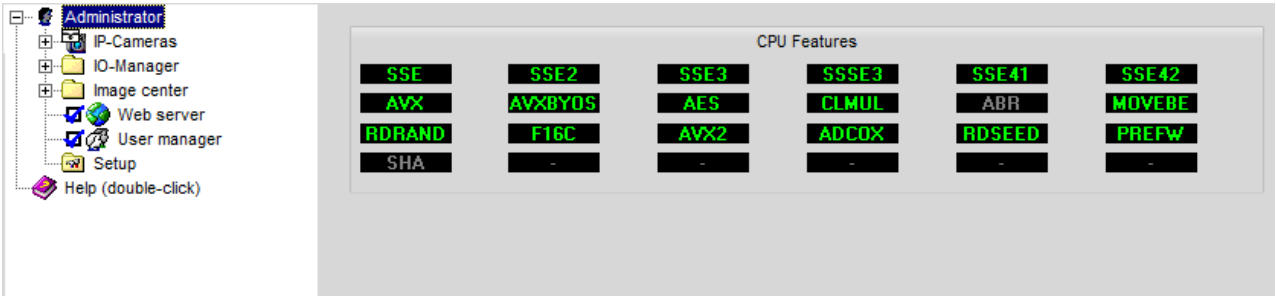
2 Processor

The faster the PC you choose for your video surveillance system, the better, in particular if you want to run a large number of cameras in parallel. But it's not only the speed, but the features of the processor used that you need to bear in mind.

Another crucial factor for achieving optimum performance is to equip the processor with relevant instruction set extensions known as SSEs (Streaming SIMD Extensions). SSE2, which has been the standard for many years, is a basic prerequisite for this. Obsolete processor families without SSE2 (e.g. AMD Athlon XP or Intel Pentium III) are therefore not supported by go1984.

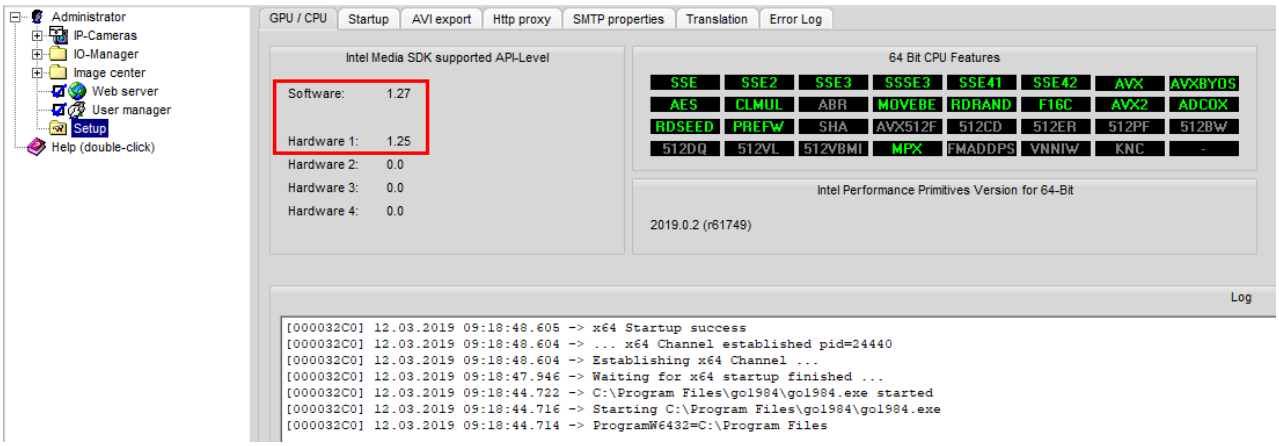
When it comes to new compression standards like H.264, instruction set extensions are vital to achieving optimum performance, with SSE4.1 as a minimum requirement. You should therefore use, as far as possible, state-of-the-art processor families, especially along with H.264 and high-resolution cameras.

The instruction set extensions supplied by the processor used are displayed for you to control in go1984:



go1984 is available for Windows 2019/2016/2012/2008/10/8.1/7. Current go1984 releases (release 7.1.0.1 or newer) offer improved performance by GPU H.264 decoding.

GPU decoding requires a 64-bit system under Windows 10 or Windows Server 2016/2019 and a current Intel Core i processor from the 6th generation (Skylake) with integrated GPU or Intel Xeon E3 v5 with Intel Processor Graphics Gen9 for media. The Intel video drivers must be up to date.

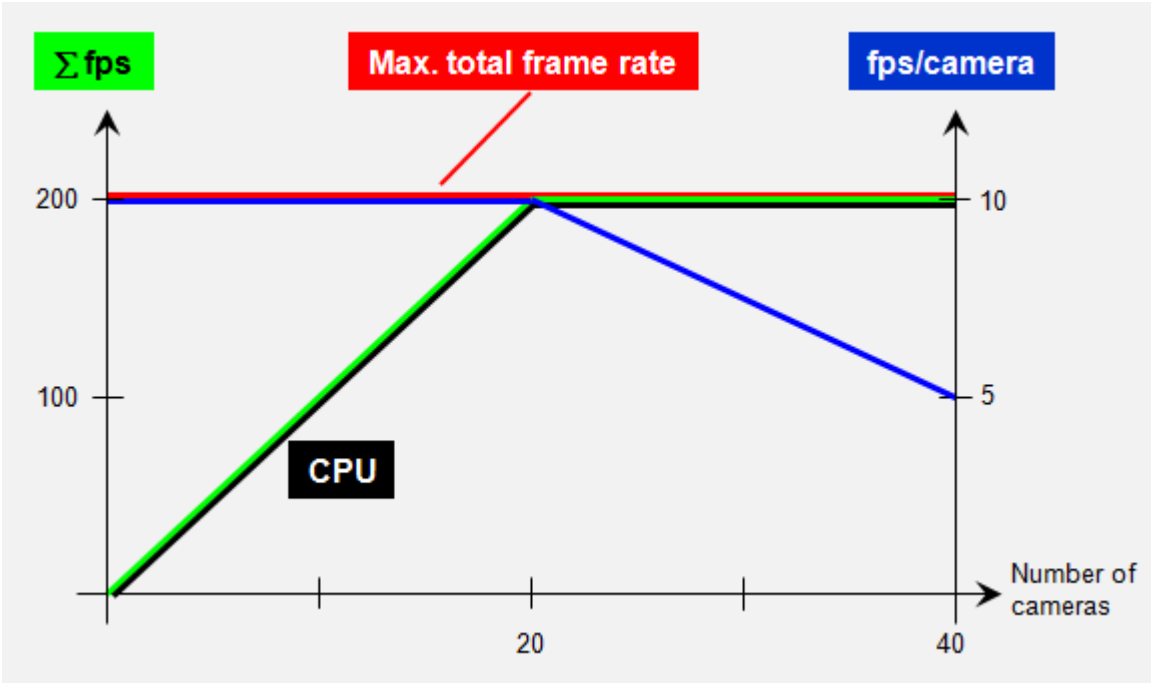


Depending on the size of the installation, we recommend, for example, the following CPUs:

go1984 Pro Edition: e.g. Core i3-7100 or Core i3-8100
go1984 Enterprise Edition: e.g. Core i5-7400 Core i5-8400
go1984 Ultimate Edition: e.g. Core i7-7700 Core i7-8700

Do not use 2- or multi-processor machines.

The maximum total frame rate will depend on the performance of the hardware. The figure below assumes a resolution of 640x480 pixels at a rate of 200 frames per second (fps). In this case the minimum rate for each camera should always, where possible, be 10 fps.



It is obvious that there is a linear increase in the CPU usage as the number of cameras grows, although the frame rate per camera remains the same (10 fps). With 20 cameras and a total of 200 fps, the PC is operating at full capacity. If more than 20 cameras are installed, the frame rate for each camera will decrease accordingly. With 40 cameras the maximum possible frame rate is only 5 fps per camera.

It is possible to run go1984 on lower performance hardware, but this will have an impact on the frame rate of each camera.

Please contact us if you have questions about your specific situation.

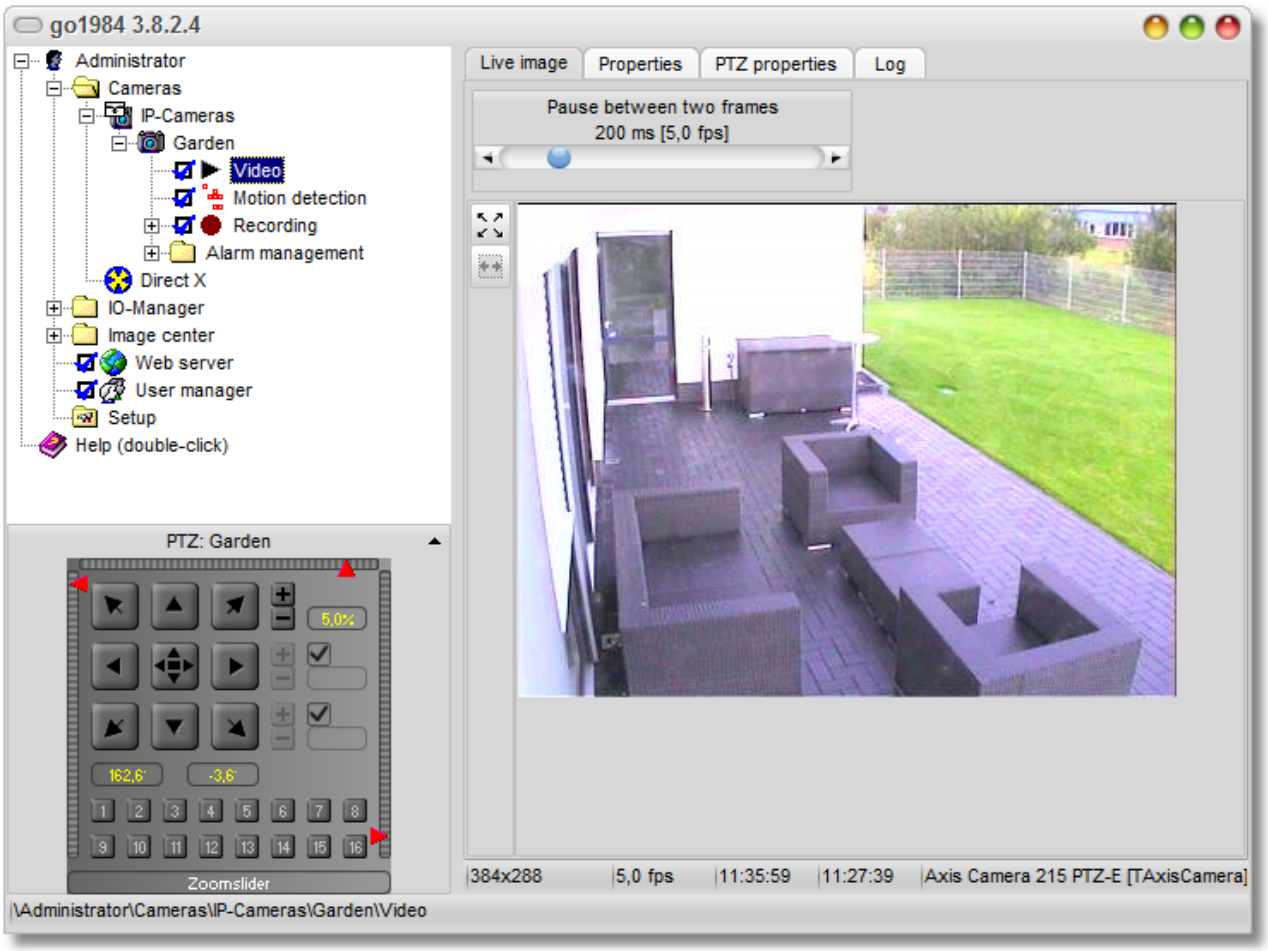
3 Pause between two frames and image size

The principle behind the choice of the pause between two frames and image size should be: as much as necessary and **not** as much as possible.

Bear in mind that higher frame rates and image resolutions will have a negative impact on the overall performance of the system and will also require significantly more disk space and bandwidth. For example, doubling the resolution (from 320x240 to 640x480) will result in images that are four times larger.

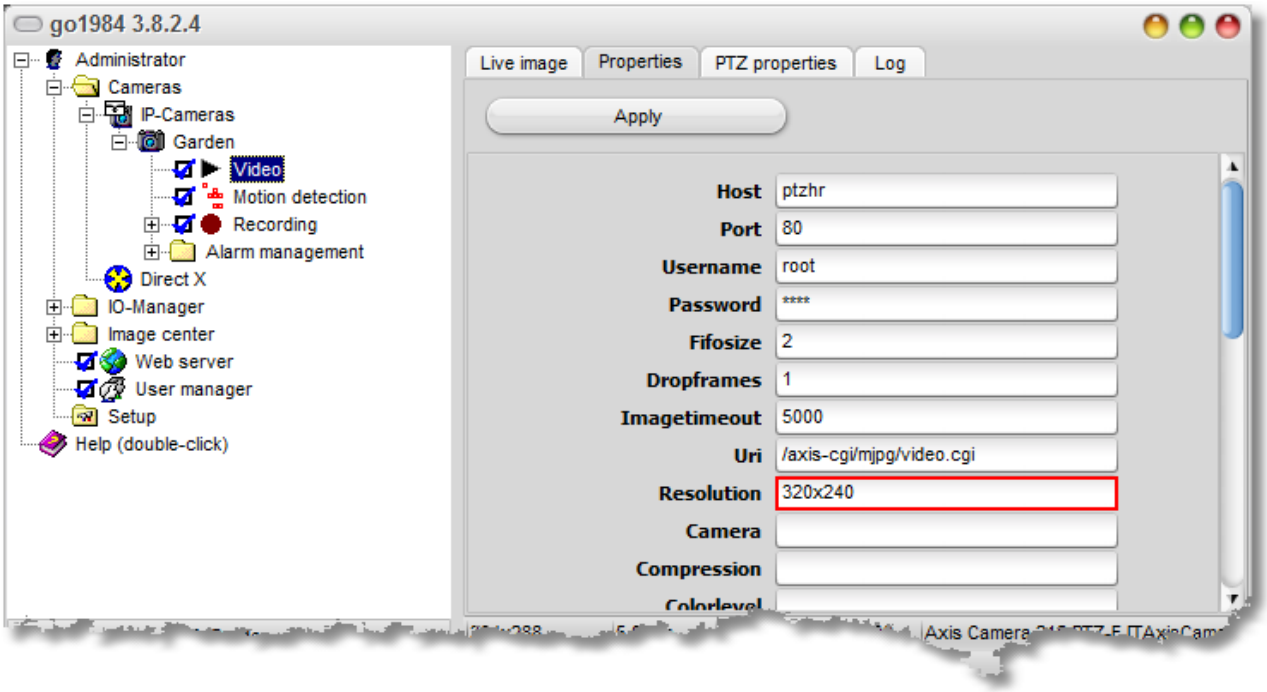
In addition, rates of between two and five frames per second are perfectly adequate in most cases. There is very little more information to be gained from higher frame rates and this will have a negative impact on performance and need considerably more disk space and bandwidth.

Set the frame rate using the "Pause between two frames" slider.



In the status bar below the live image you can see the current resolution and frame rate (fps).

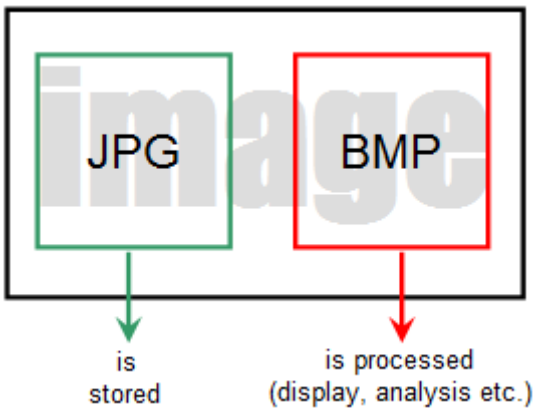
Generally the image is transferred to go1984 with the same resolution as it was given in the camera from the camera's web interface. However, some camera models allow you to enter a different setting for the image displayed in go1984.



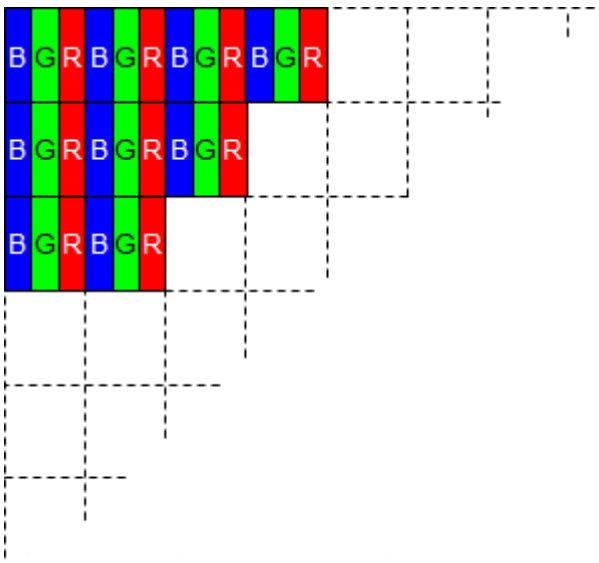
4 The fundamentals of bitmaps and JPEG files

The images supplied by the camera can be in bitmap or JPEG format, depending on the requirements and circumstances. There may be a need to convert the image from JPEG to bitmap format (decompression) or in the reverse direction from bitmap to JPEG (compression).

The dual function of a camera image:



RGB24 is a two-dimensional bitmap format. In each pixel one byte is needed for the blue, green and red channels, as shown in the figure below.



This means that each pixel consists of 3 bytes. As a result the size of an image with VGA resolution (640x480 pixels) is:

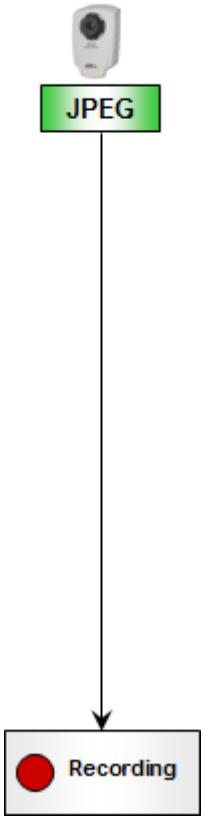
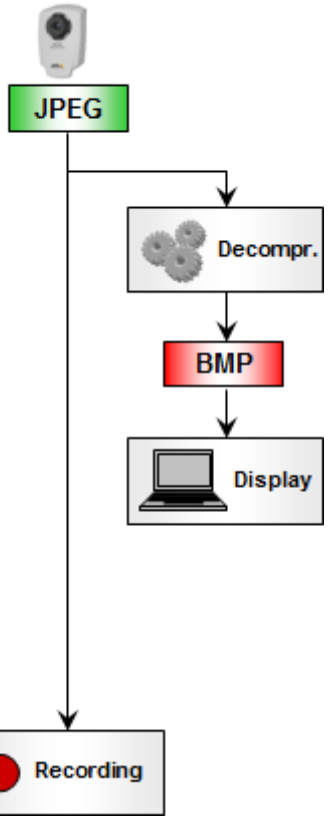
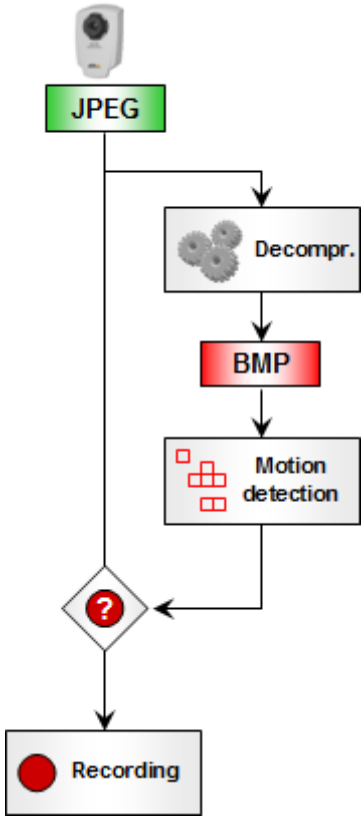
640 x 480 x 3 bytes = 921.6 KB or approximately 1 MB.

JPEG is a lossy image compression standard. We will not describe the standard in detail here. The decisive factor is that the size of a compressed JPEG image with VGA resolution is only around 30 KB, depending on the compression rate, which is approximately 3% of the size of the equivalent bitmap image.

IP cameras generally provide images in JPEG format. However, if the image has to be displayed or edited, it must be decompressed using a compute-intensive method to produce a bitmap image:

- to display the camera image (in the individual image view and central image view)
- to edit the image in go1984, for example rotating, mirroring or adding a caption
- to use active motion detection in go1984

The table below illustrates the different scenarios:

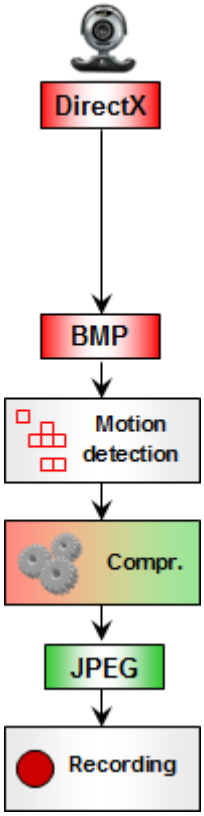
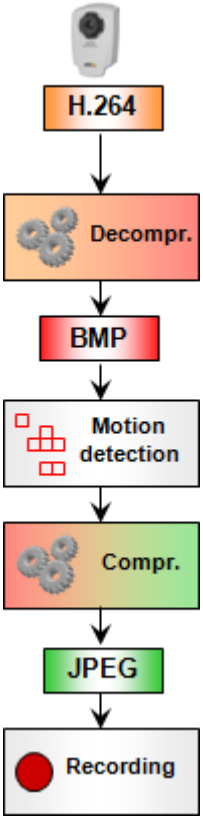
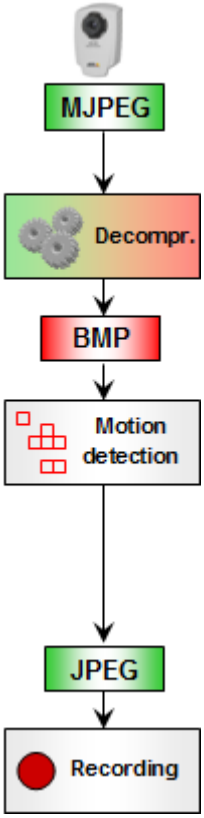
Recording without display or motion detection	Recording with display	Recording with motion detection
		

As a result, decompression should be avoided, wherever possible.

Therefore, you should close the go1984 main window if you do not need to display a live image on the go1984 PC (tray icon mode), in order to avoid the compute-intensive decompression of the image (for display purposes).
Do not display the central image view unnecessarily (for example, in the go1984 screensaver).

5 DirectX, MJPEG and H.264

The following figure explains the differences between the different image sources in the case of motion-based recording.

DirectX sources (webcam, analog camera via capture card)	H.264 sources	MJPEG sources
		
Poor Performance	***Medium Performance***	***Good Performance***
Often results in poor performance because of the camera driver and the required compression from BMP to JPEG.	Time-consuming decompression from H.264 to BMP **and** compression from BMP to JPEG necessary resulting in poor performance.	JPEG image is already available. Decompression only needed for display or motion detection, therefore generally offers the best performance.

Although no decompression is needed for DirectX sources, because the image is supplied in bitmap format, the performance of the drivers is often poor and the bandwidth needed is very high (~1 MB per VGA image).

In the case of H.264 sources, the decompression process is more time-consuming and more compute-intensive than with MJPEG sources.
Every image from DirectX or H.264 sources must be compressed into JPEG format. This requires around 1.3 times as much computing power as decompression.

Therefore we recommend the use of video servers rather than capture cards (DirectX) with analog cameras.

6 H.264

Without any knowledge of its pros and cons, H.264 is often described as the “ultimate” in image transfer when used with digital video surveillance solutions.

Compared with its predecessors, it is actually a high-quality compression standard that allows you to transfer large volumes of data at a comparatively low bandwidth. The fewer the changes that occur to each individual image (i.e. the less movement there is in front of the lens), the more effective the result achieved through image compression. In general terms, we are talking about using around 20% of the bandwidth required, compared with the MJPEG format.

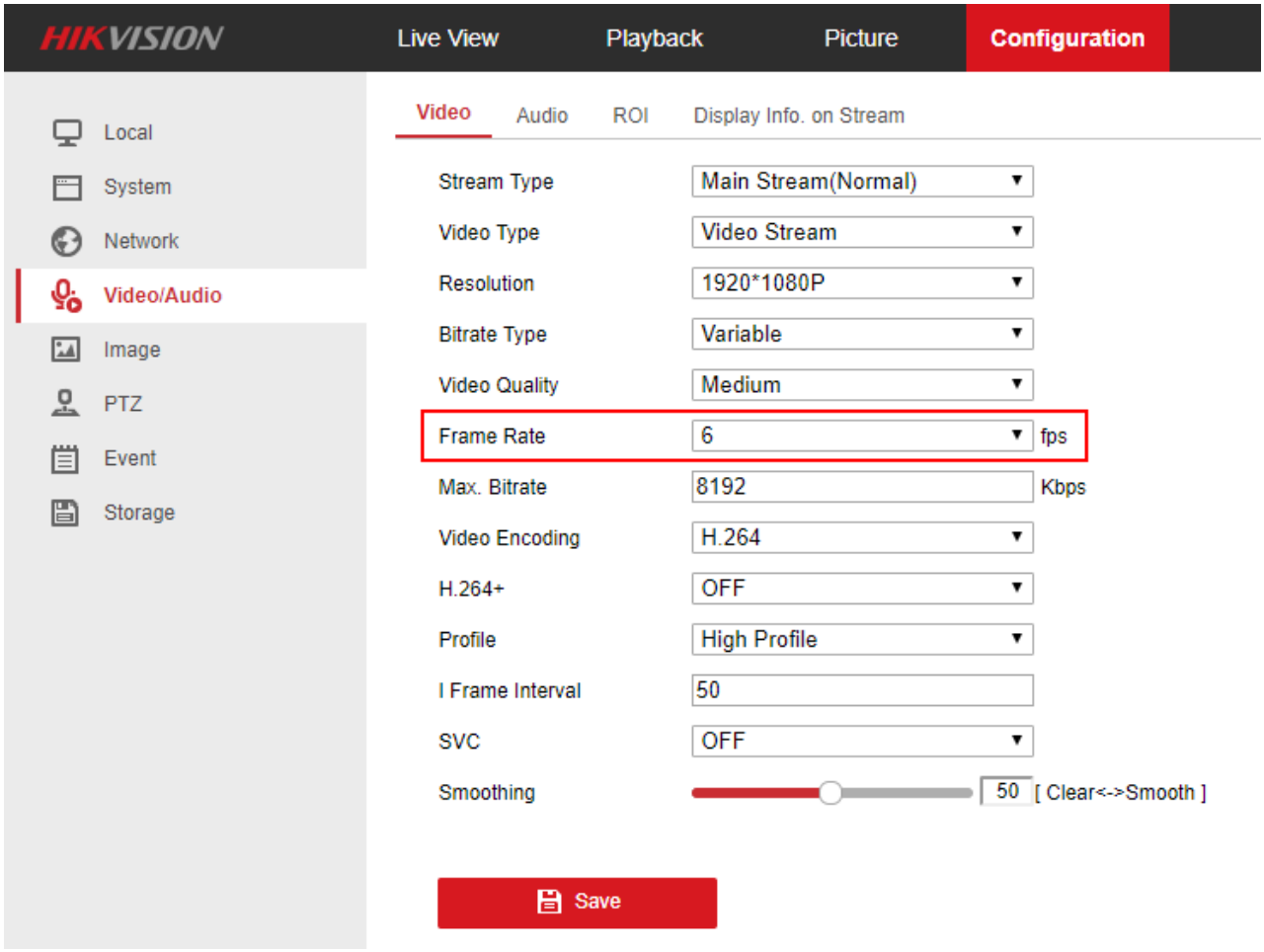
However, all too often, no mention is made of the processing time required to decompress the video stream.

Indeed, go1984 uses current and optimized Intel libraries for decompressing the images, but this is still, by comparison, significantly more time-consuming than decompressing JPEG images. In addition to this, the images also need to be compressed to JPEG format. As has already been mentioned, this compression operation sometimes requires approx. 1.3 times additional processing time compared with decompression.

If you are using H.264 compression, state-of-the-art processors with integrated GPU should definitely be selected. GPU decoding requires a 64-bit system under Windows 10 or Windows Server 2016/2019 and a current Intel Core i processor from the 6th generation (Skylake) with integrated GPU or Intel Xeon E3 v5 with Intel Processor Graphics Gen9 for media.

You should also always match the refresh rate at camera level with the refresh rate set in go1984. This means that if you choose “5 fps” as the default refresh rate in go1984, set the camera’s rate to “5 fps” as well. If this synchronization does not happen, images will have to be needlessly decompressed, which can sometimes have a significant impact on overall performance.

Example: Setting the refresh rate (“Frame Rate”) for a Hikvision IP camera:



If sufficient bandwidth is available, we recommend you use MJPEG cameras or retrieve the MJPEG stream from the cameras.

For example, 30 MJPEG cameras, each with a refresh rate of 10 fps in a gigabit Ethernet network, generate a load of approx. 7.7% (30 x 10 fps x 32 kByte = 9,600 kByte/s = 76,800 Kbps = 77 Mbps), whereas, in H.264 mode, based on an average movement intensity of approx. 20% of this, a network load of approx. 1.5% should be expected.
Comparing the data transfer rates in this example should hardly present a convincing argument in favor of H.264 when comparing the CPU load at the same time.

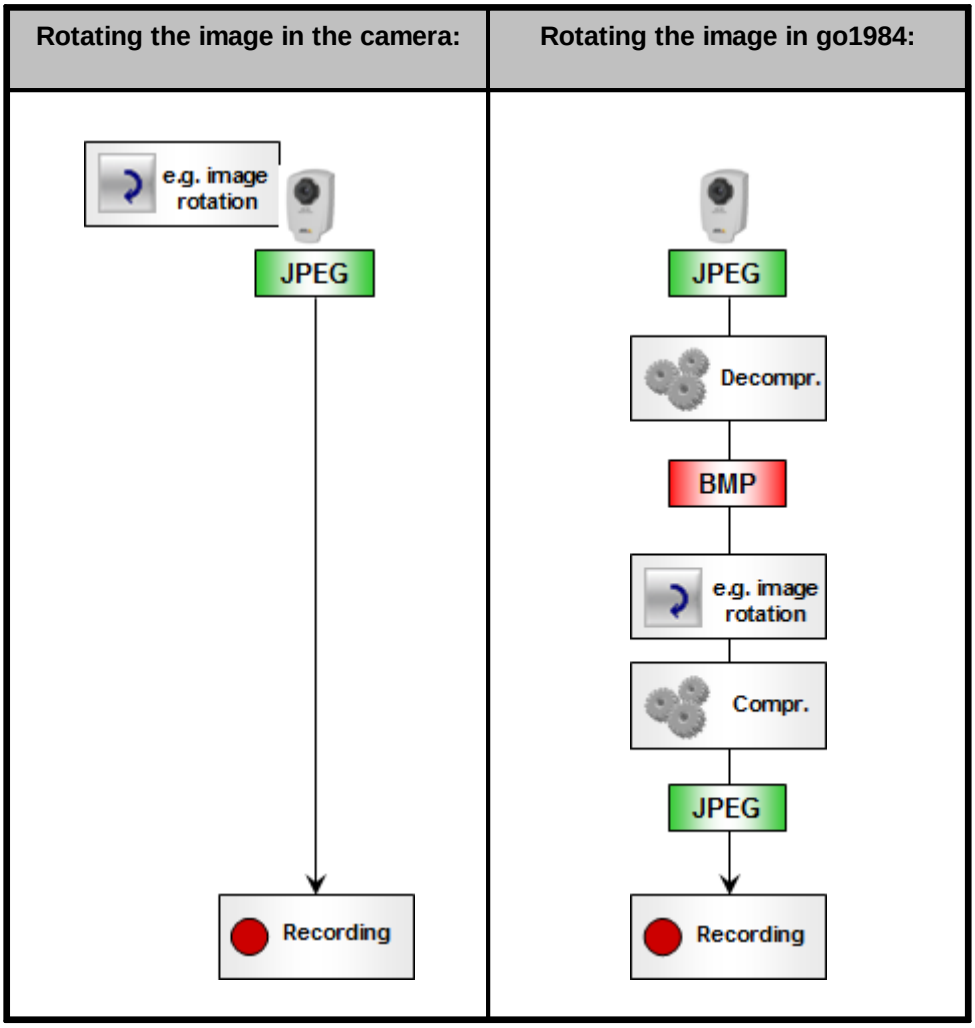
This means that it should make sense sometimes to use it only when a narrow-band connection is used (e.g. connecting remote IP cameras via a standard DSL cable).

Note on this point that depending on the edition H.264 streams are possibly scaled down:

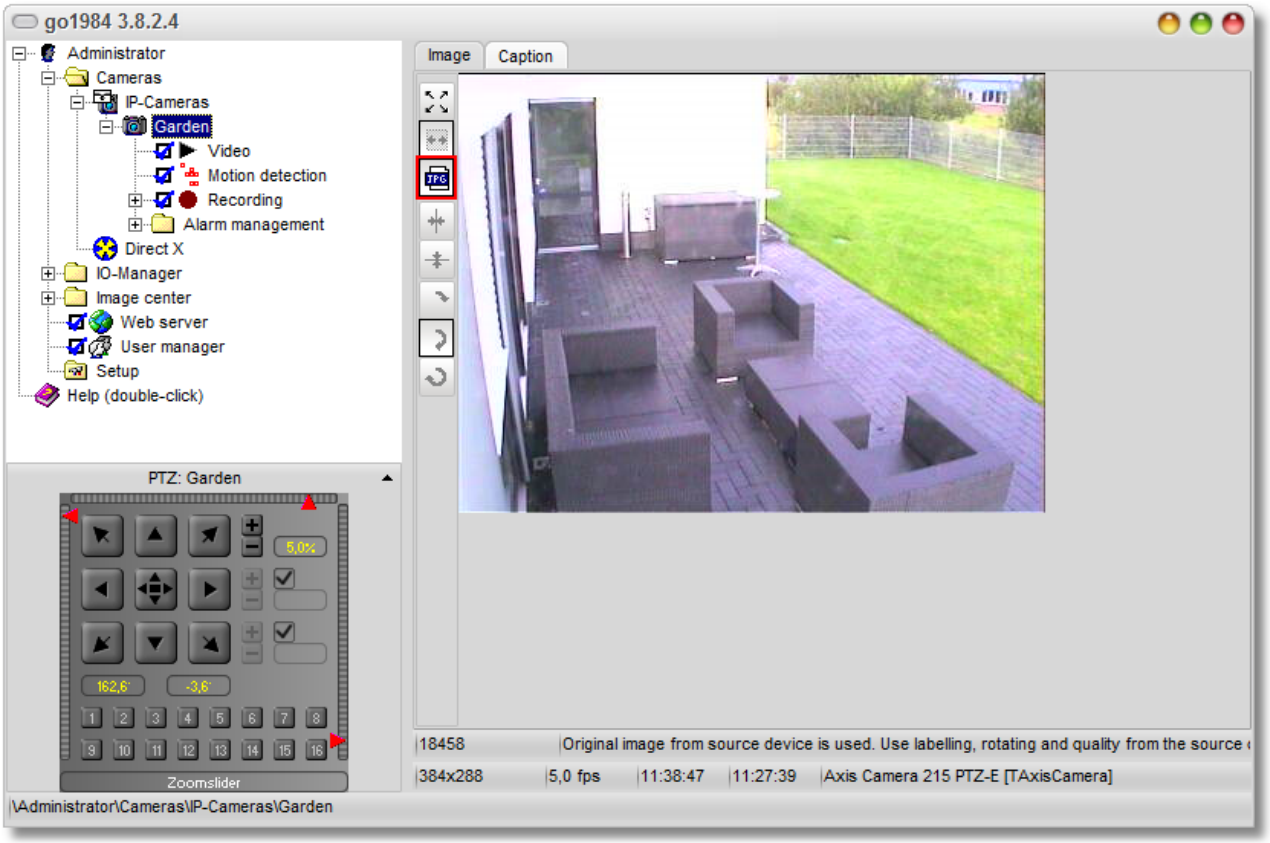
Feature	Pro	Enterprise	Ultimate
Processing of H.264 up to max. resolution	1920x1080	2688x1520	3840x2160

7 Using the original image

Wherever possible you should use the original JPEG image from the IP camera in go1984. The image can then be transferred one-to-one, without needing to be decompressed and compressed, which is highly compute-intensive. The following example illustrates this:



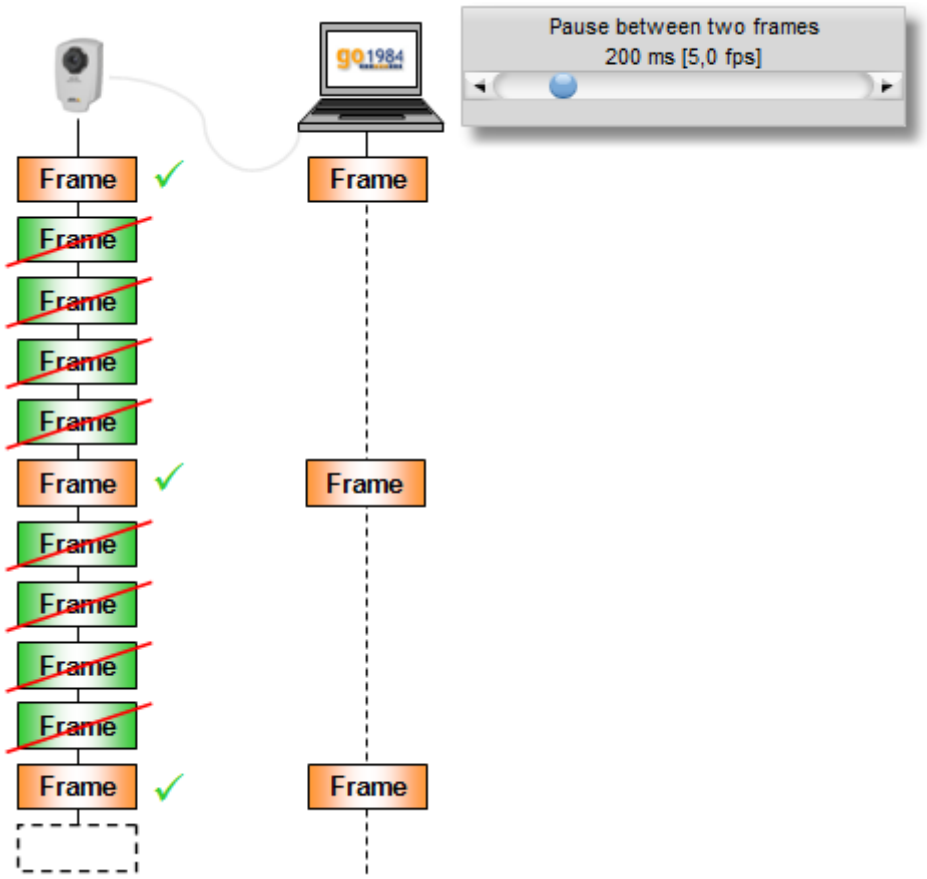
Click on the JPG button in go1984 to use the original image from the recording source (this is the default setting). Although it is no longer possible to edit the image, for example rotating, mirroring or adding a caption, this can almost always be done by the camera, which results in a significant improvement in the overall system performance.



8 Making best use of MJPEG sources

After you have connected the PC running go1984 with the MJPEG camera, the images will start arriving from the camera. If you do not set the required frame rate at the camera level, the maximum number of images will be supplied, regardless of the setting in go1984.

Superfluous frames will then be dropped, which places an unnecessary load on the network and the PC, as the following figure shows. In this case go1984 fetches an image every 200 ms, while the camera makes five images available during this time.

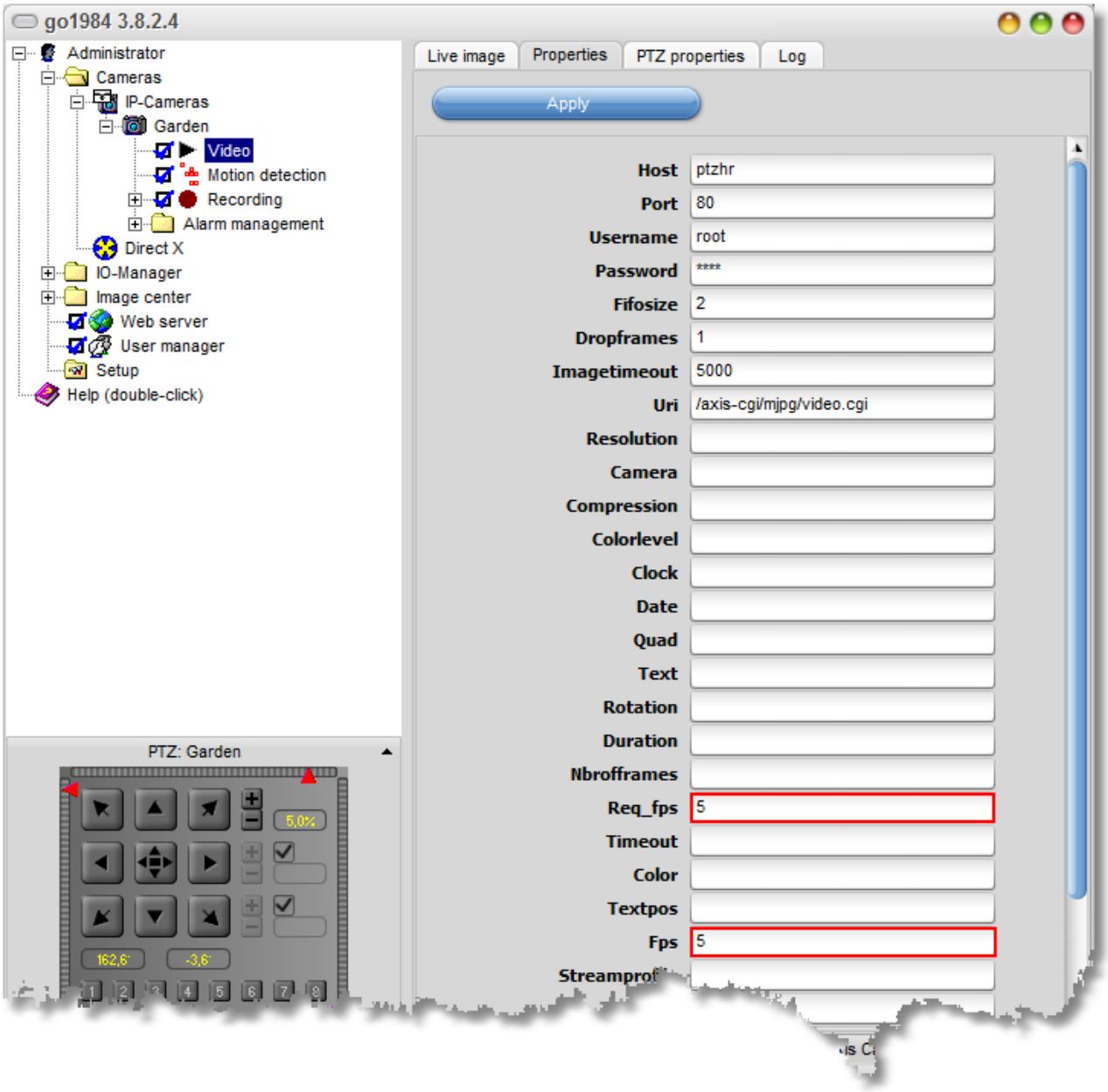


This approach is necessary in go1984 in order to guarantee that the images create a live impression, without a time lag.

Therefore, when using MJPEG cameras you should not only set the required frame rate, wherever possible, in go1984, but you should also set the maximum frame rate at the camera level and ensure that the two rates are the same.

It is worth mentioning that this approach is not needed with JPEG cameras, as they use a handshake at http level.

In the case of Axis cameras, you should set the required frame rate (in this case 5) in the fields Req_fps and Fps in the Video->Properties tab, in the form of a whole number with no decimal places. These two parameters access the same value, but may have different names depending on the firmware.



9 Practical example

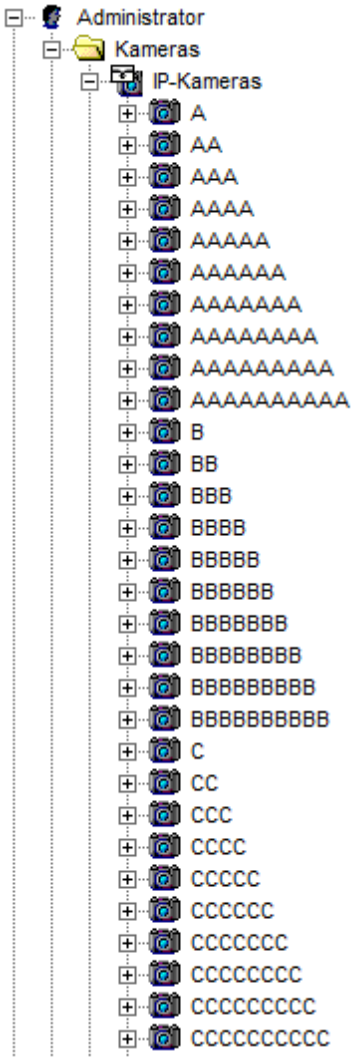
The following example should clearly highlight the differences between achieving a poor performance and an excellent performance using a system configured with both H.264 and MJPEG cameras.

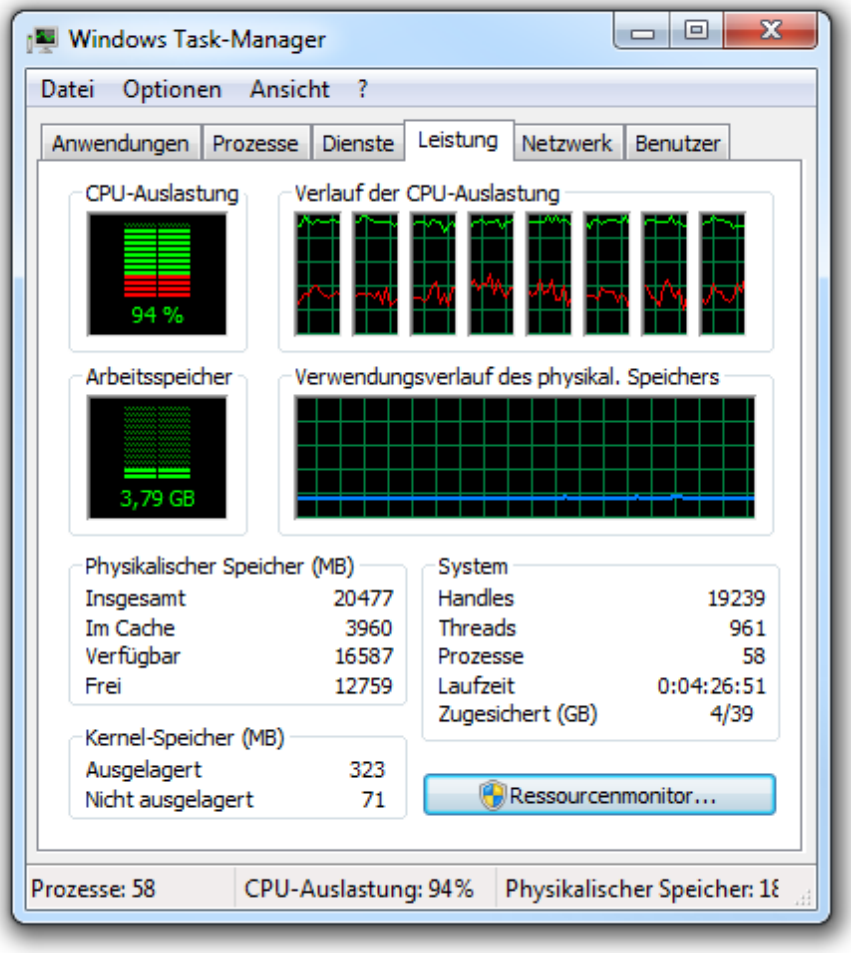
A total of 30 network cameras are connected, which record when any movement is detected. The live picture does not need to be displayed for this. There are 30 axis cameras with a VGA resolution (640x480 pixels), which each have an image record pause set in go1984 of 100 ms (10 fps), i.e. 300 fps in total.

The PC has a dual Intel Xeon E5420 2.5 GHz quad-core processor CPU with 20 GB RAM. It runs Windows 7 Ultimate 64-bit SP1 as its operating system. It also runs go1984 Ultimate Edition V3.8.2.4..

A1) Poor settings for H.264 cameras

- the max. frame rate of 10 fps was not specifically set at camera level
- motion detection involves every single image being analyzed
- due to H.264 mode being used, the time-consuming H.264 -> BMP decompression operation is required, along with an additional BMP -> JPEG compression operation for each single image

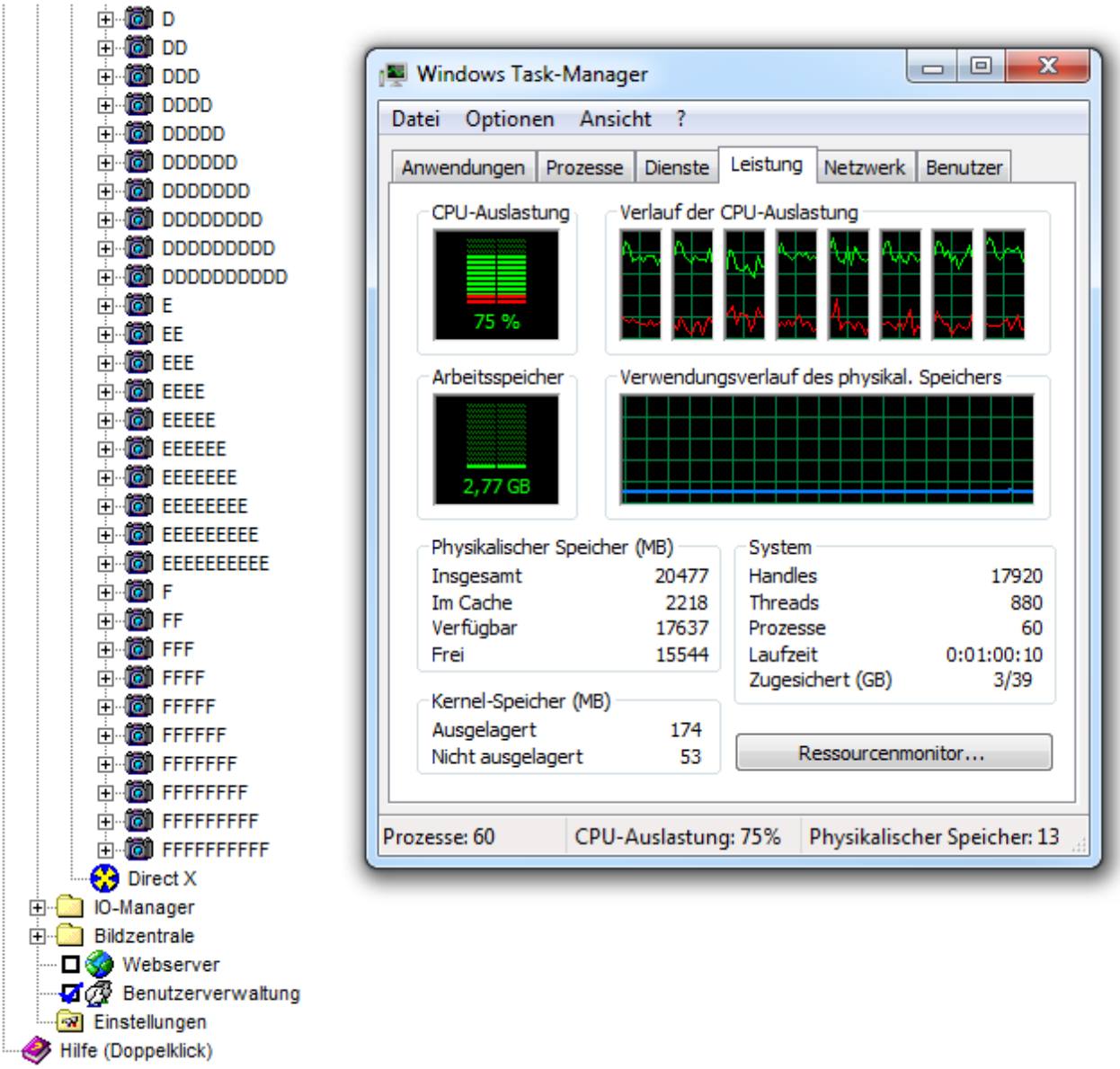




The **CPU load is almost 100%**, which means that, in total, an overall refresh rate of just around **150 fps** will be achieved instead of the required rate of 300 fps.

A2) Poor settings for MJPEG cameras

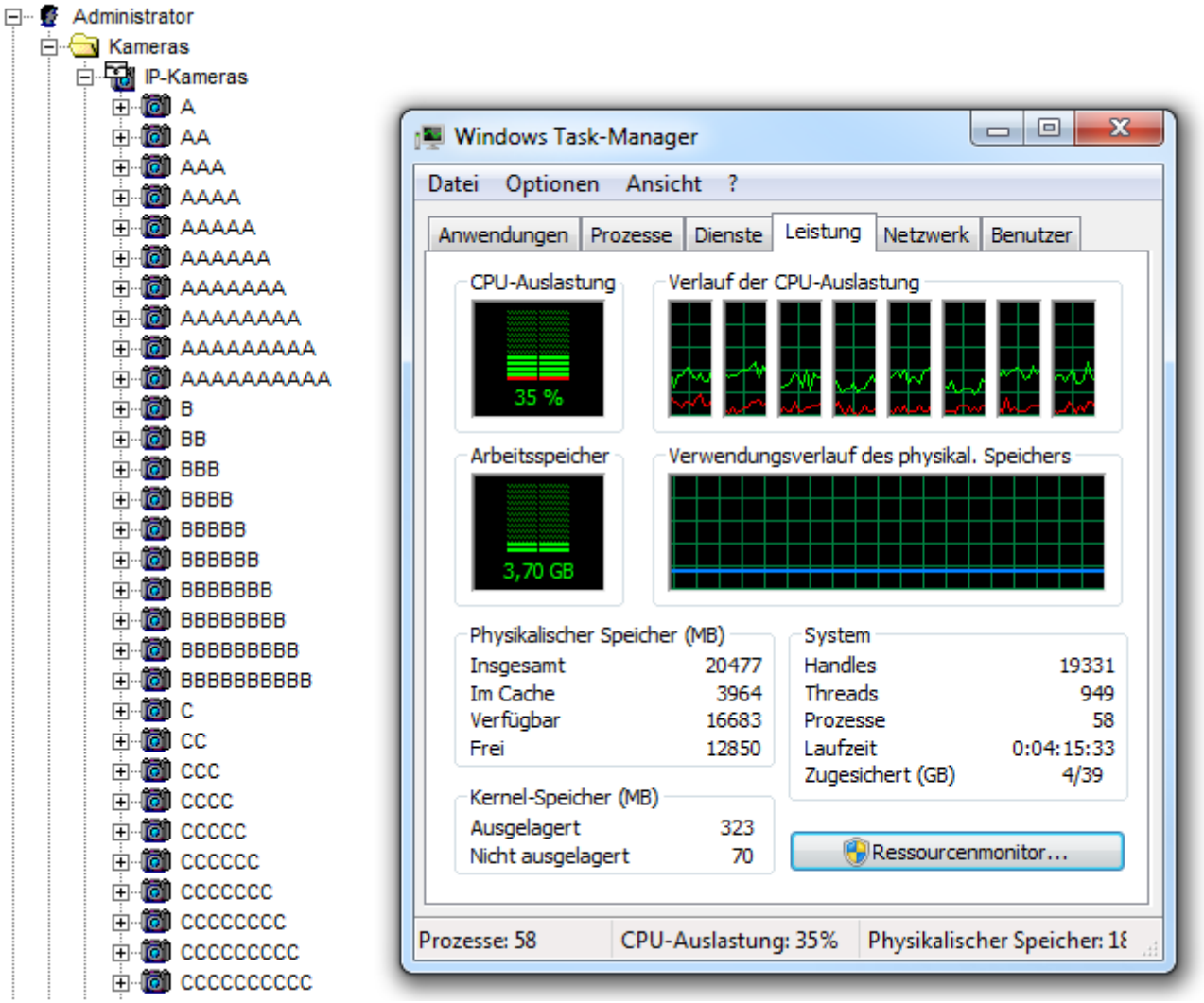
- the max. frame rate of 10 fps was not specifically set at camera level
- motion detection involves every single image being analyzed and therefore decompressed
- image rotation in go1984 and, apart from decompression, a time-consuming compression operation needs to be carried out on each image from bitmap to JPEG.



The **CPU load is 75%**, which means, in total, that an overall refresh rate of approx. **275 fps** is achieved.

B1) Optimized settings for H.264 cameras

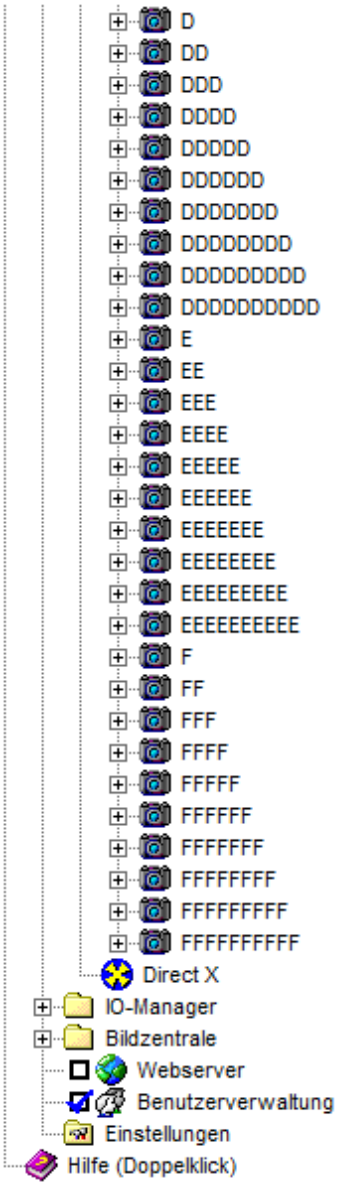
- frame rate set at camera level to 10 fps for each camera
- only one camera image in five (two images a second) is analyzed
- due to H.264 mode being used, the time-consuming H.264 -> BMP decompression operation is still required, along with an additional BMP -> JPEG compression operation for each single image

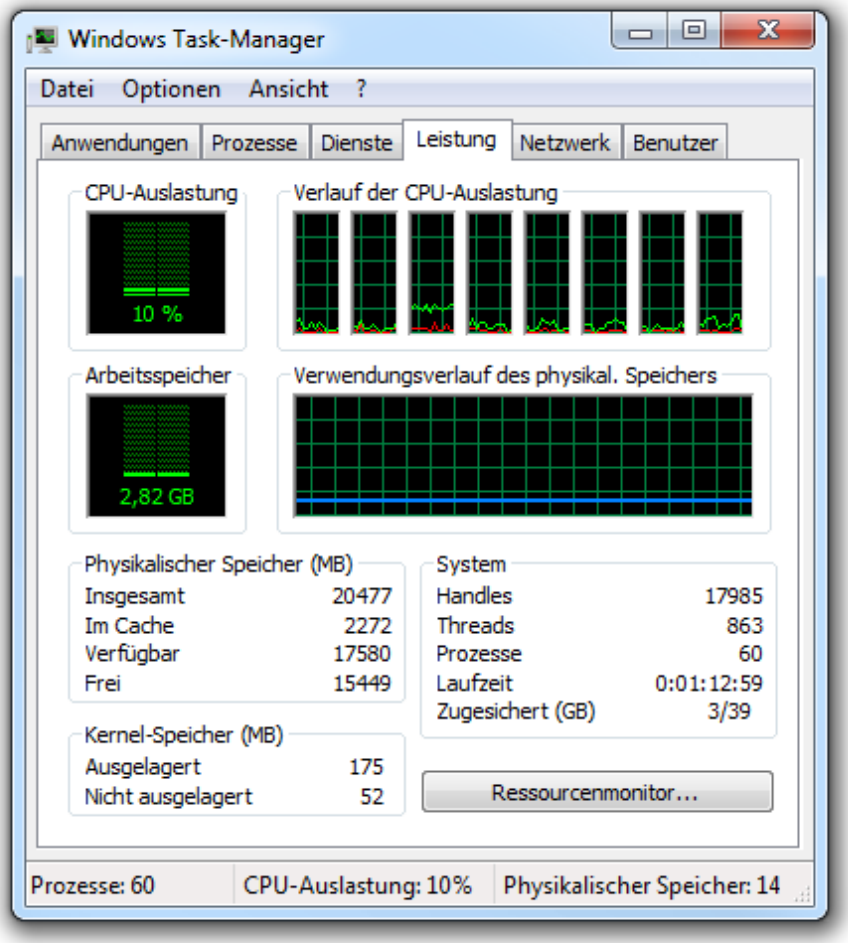


The **CPU load is 35%**, which means, in total, that the overall refresh rate setting of **300 fps** is achieved.

B2) Optimized settings for MJPEG cameras

- frame rate set at camera level to 10 fps for each camera
- only one camera image in five (two images a second) is analyzed and decompressed
- image rotation is carried out directly in the cameras, which means the bitmap-to-JPEG compression operation is no longer required





The **CPU load is still only approx. 10%**, which means, in total, that the **300 fps** setting is easily achieved.

The following table shows the summary of the test results:

	CPU Load as %	achievable overall frame rate in fps
A1) H.264 cameras with poor settings	**100**	**150**
A2) MJPEG cameras with poor settings	75	275
B1) H.264 cameras with optimized settings	35	300
B2) MJPEG cameras with optimized settings	**10**	**300**

The system configured for H.264 cameras producing a poor performance only achieves a rate of 150 fps with a full CPU load. A system configured for MJPEG cameras producing a comparably poor performance can achieve at least double the frame rate at a 25% lower CPU load.

The optimized configuration generates a CPU load of 35% in H.264 mode, whereas a similarly configured system for MJPEG cameras generates only approx. a third of this CPU load, i.e. 10%.

10 Benchmark tests without remote access

We performed suitable benchmark tests for you so as to differentiate between the various go1984 editions available in terms of general performance and performance-related limits.

The tests were carried out using go1984 version 3.8.2.4 on 80 axis M1011 cameras with a VGA resolution (640x480 pixels), operated in MJPEG mode. One image in five was analyzed in motion detection mode, while any movements detected were recorded.

Dual CPU Intel Xeon E5420 (8 Cores) with 2.50 GHz each, 20.00 GB RAM
Windows 7 Ultimate SP1 64-bit

Edition	Pro	Enterprise	Ultimate
max. overall fps	180	360	730
CPU utilization in %	10	15	20
RAM utilization in MB	250	260	280

AMD Phenom X4 9750 Quad-Core 2.40 GHz, 4.00 GB RAM
Windows Server 2008 Enterprise 64-bit

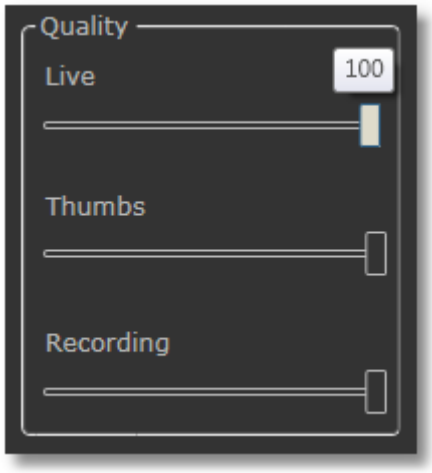
Edition	Pro	Enterprise	Ultimate
max. overall fps	128	250	500
CPU utilization in %	20	25	30
RAM utilization in MB	235	240	250

The benchmark tests show that go1984 Enterprise Edition and go1984 Ultimate Edition can process images at a refresh rate twice and four times respectively that of go1984 Pro Edition. It is also evident that multi-core processors are extremely beneficial for go1984. With a VGA resolution, rates of approx. 25 fps per core can be achieved with go1984 Pro Edition, 50 fps per core with go1984 Enterprise Edition and approx. 100 fps per core with go1984 Ultimate Edition.

11 Benchmark tests with remote access

Active go1984 web server sessions will obviously increase the load. If you use a browser to access the go1984 web server via a LAN, you should choose a high bandwidth in order to avoid the server needing to recompress the images.

This also applies to access via go1984 Desktop Client or go1984 Web Client, where you can use the appropriate sliders to request the images in original quality:

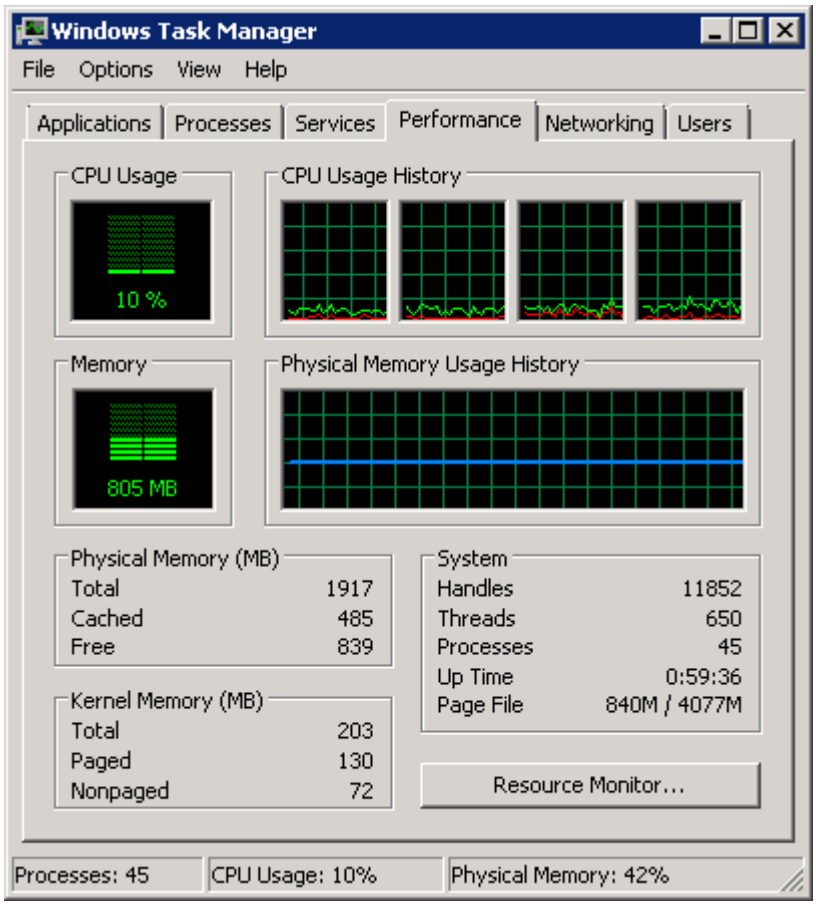


The go1984 Desktop Client also offers a motion mode. In live mode images are permanently transferred, regardless of whether any movement is detected. In motion detection mode images are transferred depending on whether any movement is made. This means that images are updated only when movement is detected by the relevant camera.

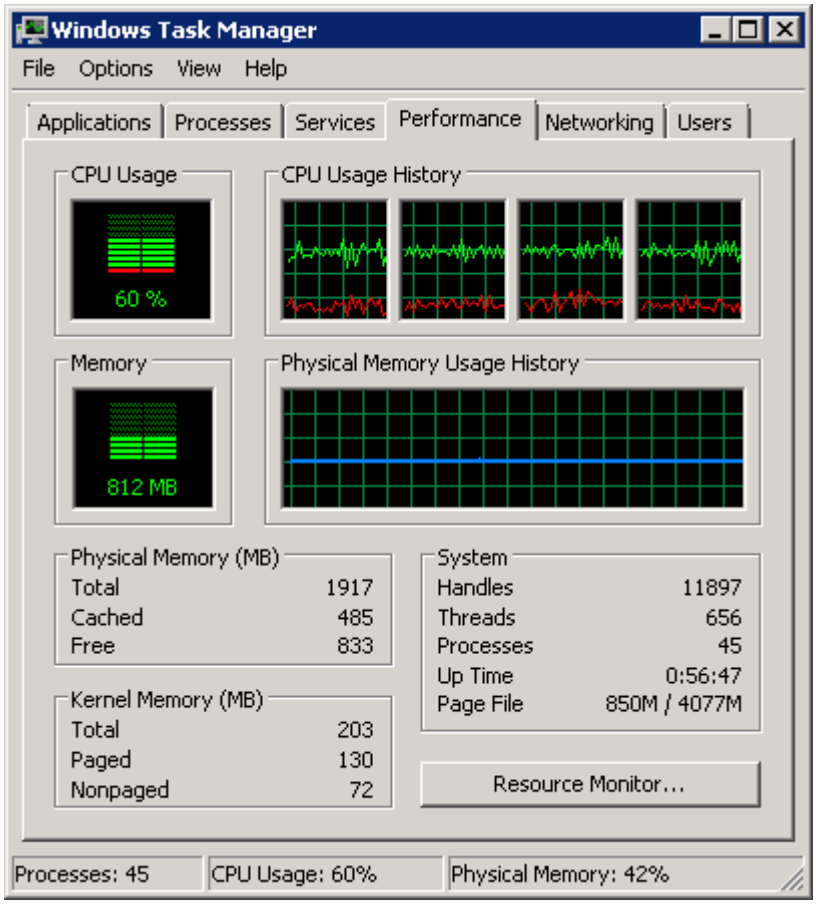
Motion detection mode is generally recommended so that you can ensure that the best use is made of the bandwidth and optimum performance can be achieved both on the server and client side.

In a web server benchmark test we operated on the server side 16 Axis MJPEG cameras with a VGA resolution, each with a refresh rate of 10 fps. The web server accessed two client PCs in parallel via go1984 Desktop Client. The cameras were set to display a live 4x4 view. The server PC was an AMD Phenom X4 9750 2.4 GHz quad-core with 4 GB RAM, running Windows Server 2008 Enterprise 64-bit and go1984 version 3.8.2.4.

The CPU load on the server side without any active web server sessions was approx. 5%. With two active clients (16 cameras x 10 fps x 2 clients = 320 fps in total) this figure rose to approx. 10% if the images had to be displayed in their original quality.



If the images had to be compressed, which meant that they had to be recompressed on the server side, the CPU load increased six-fold to 60%!



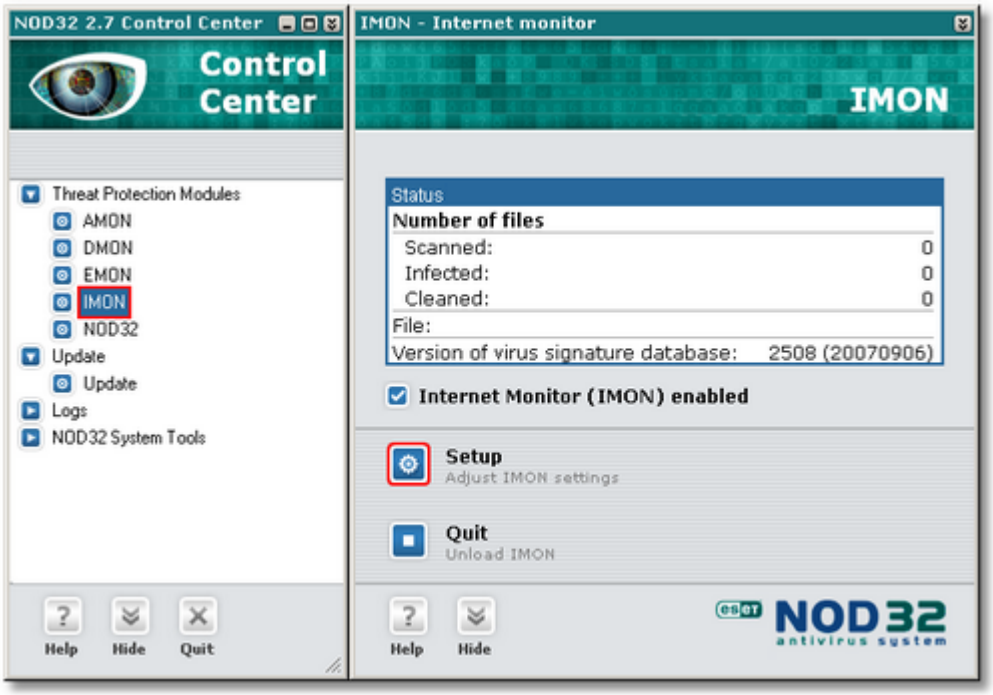
12 Impact of antivirus software

Remember that in some cases, antivirus software can also have an adverse influence on your system's performance.

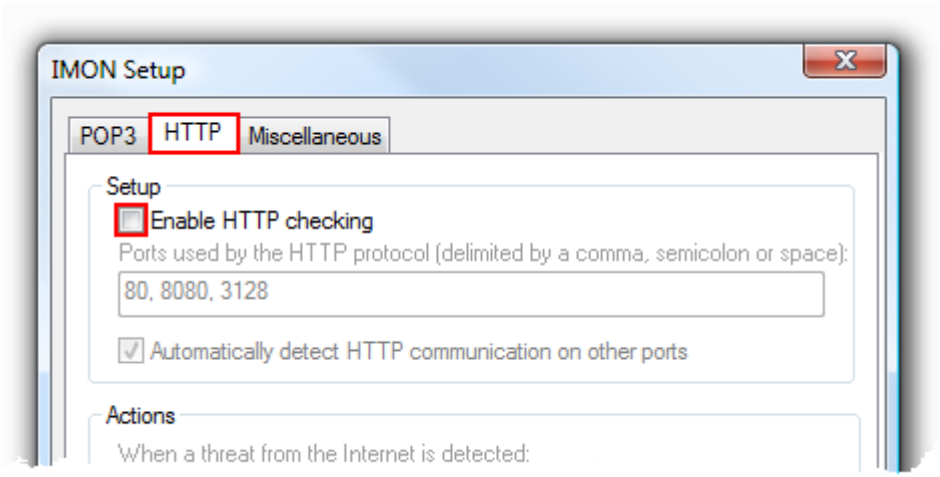
The load exerted by antivirus software on the CPU can sometimes be displayed in the Task Manager at the go1984 process's expense (e.g. using a DLL injection). In case of doubt, the best thing to do is to check your system's load with and without active antivirus software in order to detect possible effects and enter, if possible, the go1984 process and the folder(s) for the recordings as exceptions.

For example, a known problem relating to "NOD32" should be listed here. The default active HTTP check results in the connections to the cameras being constantly analyzed. Apart from the adverse impact on performance, the hard disk is populated with log files in next to no time. This check does not make much sense particularly at LAN level and can be deactivated as described below.

Open the "Control Center" and select "IMON" -> "Setup".



Then select the "HTTP" tab and deactivate the HTTP check by unchecking the option "Enable HTTP checking".

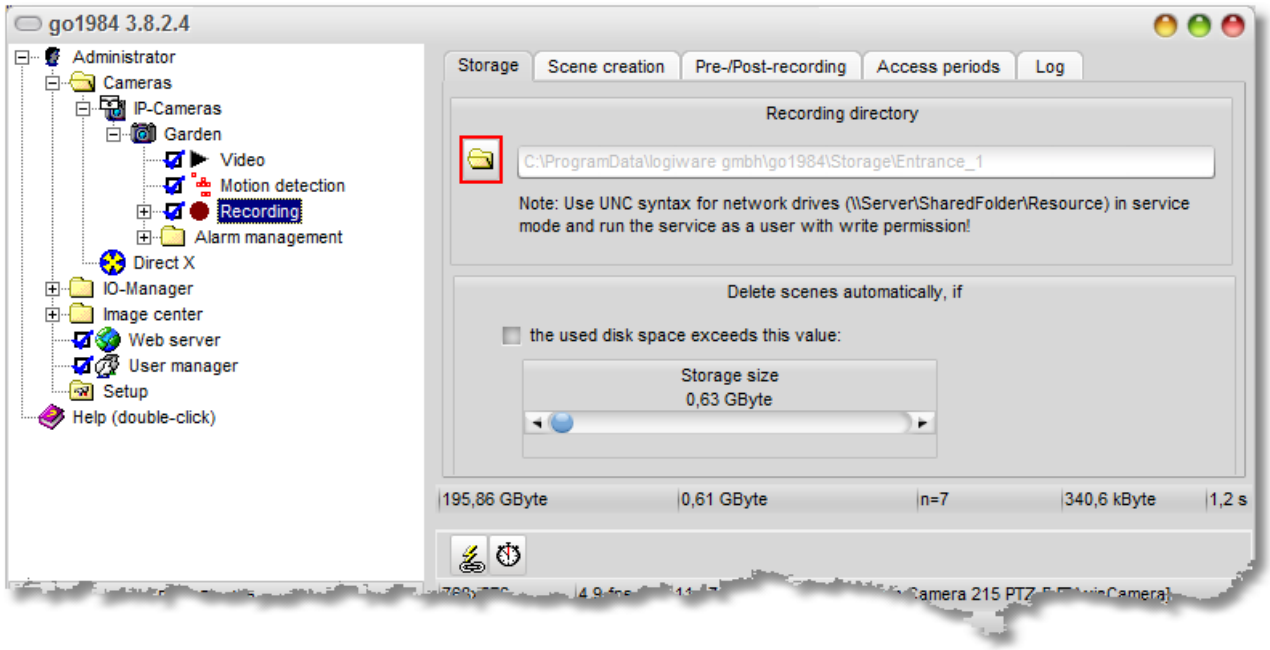


You should check beforehand whether it is advisable to adapt the settings on your system for security reasons.

13 Using several hard disks

Using a large number of cameras will require, during active recording, multiple parallel write access operations to be carried out to the hard disk(s). If recordings are viewed on the go1984 computer or even every time the web server is accessed concurrently by several users, multiple parallel read access operations are also required.

In these circumstances, it is generally recommended that the recordings are distributed across several local hard disks from a performance perspective. The path for each camera can be defined in go1984.



We will not go into detail here on the numerous techniques used for operating two or more hard disks simultaneously (RAID) or on the individual pros and cons of this. You are referred to the relevant literature and Internet sources on this subject.

14 Editorial information

logiware GmbH
Alfred-Mozer-Str. 51
D-48527 Nordhorn
Germany

Tel: +49-5921-7139925
Fax: +49-5921-7139929

Website: http://www.go1984.com
Sales e-mail address: sales@logiware.de
Support e-mail address: support@logiware.de