Multi-Core Acquisition Optimizer (MAO)

Introduction

Industrial machine vision systems typically use off-the-shelf computers to process images acquired from industrial cameras. Standard operating systems assign network communications exclusively to one CPU core. For a person working in an office setting using programs that do not use the network and the CPU intensively at the same time, this is fine.

But in the image processing world, it's a different picture. One or several cameras are sending image data to a single computer. The amount of data can greatly increase in multi-camera, high-resolution, or high-speed applications. The computer handles the network communication and also all of the image processing. While a computer will assign image processing work to every CPU core, the single-CPU limitation on network communication can lead to network data losses if that CPU becomes overloaded by other work.

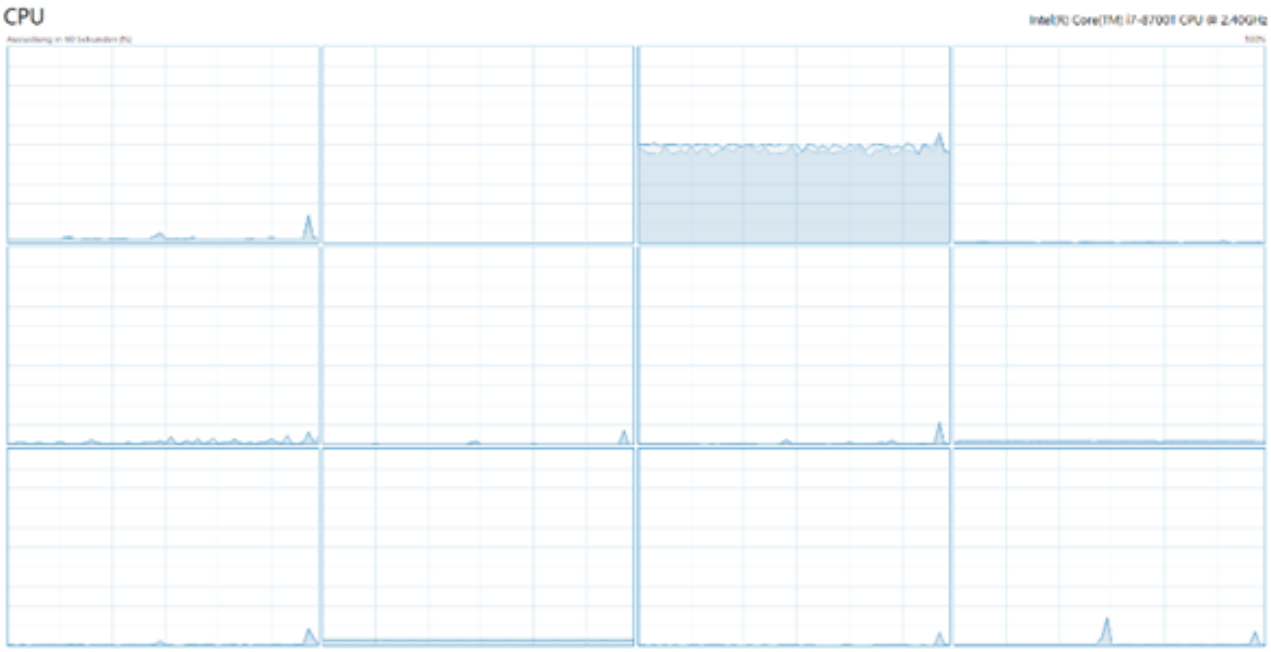
This document shows the network data processing problem and how you fix this with the Multi-Core Acquisition Optimizer (MAO) together with GigE Vision cameras of the following families:

- mvBlueCOUGAR-XT - 10Gigabit Ethernet camera series
- mvBlueCOUGAR-XD - Dual Gigabit Ethernet camera series
- mvBlueCOUGAR-X - Gigabit Ethernet camera series

Initial Situation

A network connection related to the operating system is always processed by a certain CPU core. This procedure works well and improves the system performance if several parallel connections from one device transmit similarly low volumes of data per time window. However, with GigE Vision devices a single network connection contains all or at least a significant amount of the data that are received from a network card (NIC) in the system. As a result, one CPU core is overloaded with more work than another.

With a 1 GBit/s connection the optimized NIC drivers and GigE Vision filter drivers in combination with a high-performance CPU can manage the incoming data. However, 5 GBit/s or higher bandwidths, which come from one device via a single connection, pose a much greater challenge for current CPU architectures.



MATRIX VISION GmbH · Talstraße 16 · 71570 Oppenweiler · Tel. +49-7191-94 32-0 · Fax +49-7191-94 32-288 · info@matrix-vision.de

Subject to change without notice. Date 05/2022

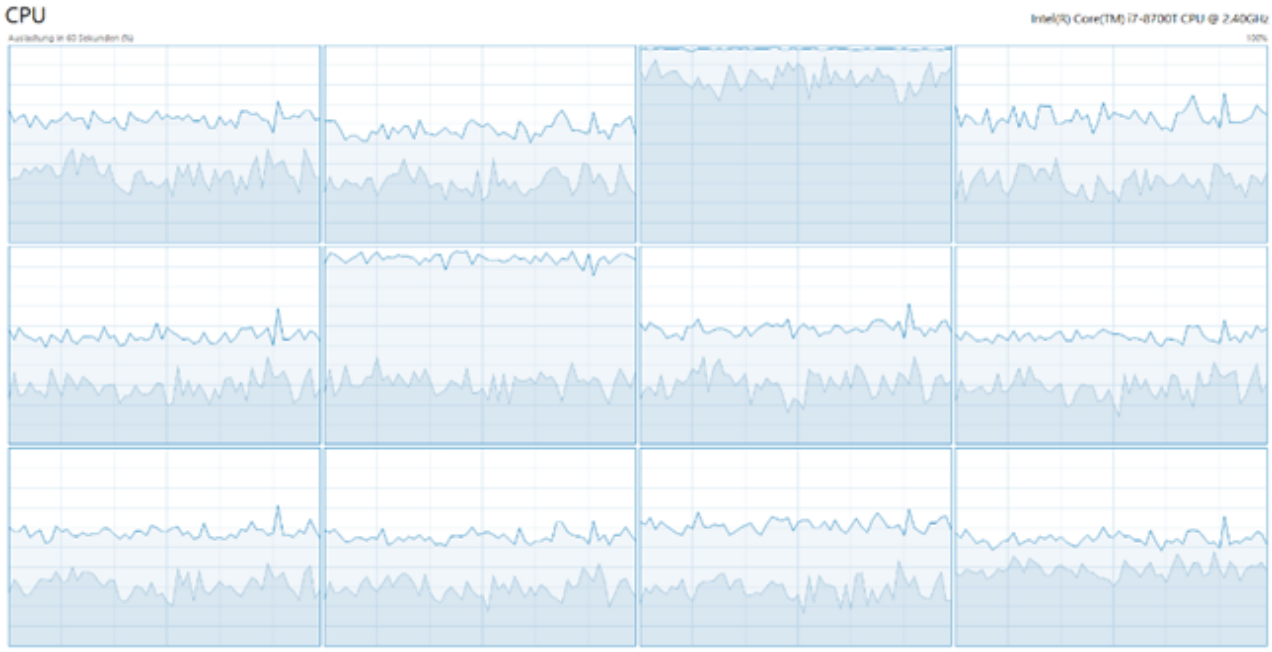
We Change Your Vision.

www.matrix-vision.de

1st caption: One core is occupied with data transmission with a data throughput of 1245 MB/s.

Figure 1 shows that the CPU load generated by a 10GigE device is relatively high. **Even though only one core is occupied with the data transmission, additional tasks may cause a data loss.**

If a CPU-intensive algorithm such as De-Bayering, Fourier Transformation, or large matrix multiplications are also to be run on the system, then the operating system of this algorithm assigns work packages of the same size to each CPU core. Also to the CPU core which already processes the incoming network data. This can lead to an overload of this CPU core and thus data losses. Furthermore, the other cores are not optimally utilized as the maximum size of the additionally possible work packages are limited by the already high workload of that one core.



2nd caption: A CPU-intensive algorithm is run on the PC in addition to the data stream. The workload of this algorithm is also distributed to all cores.

Even if the core is not overloaded, problems may occur given the fact that the processor must execute tasks for a short period of time that are not connected with the processing of the network data. If the internal buffer overflows as a result, this leads to a data loss.

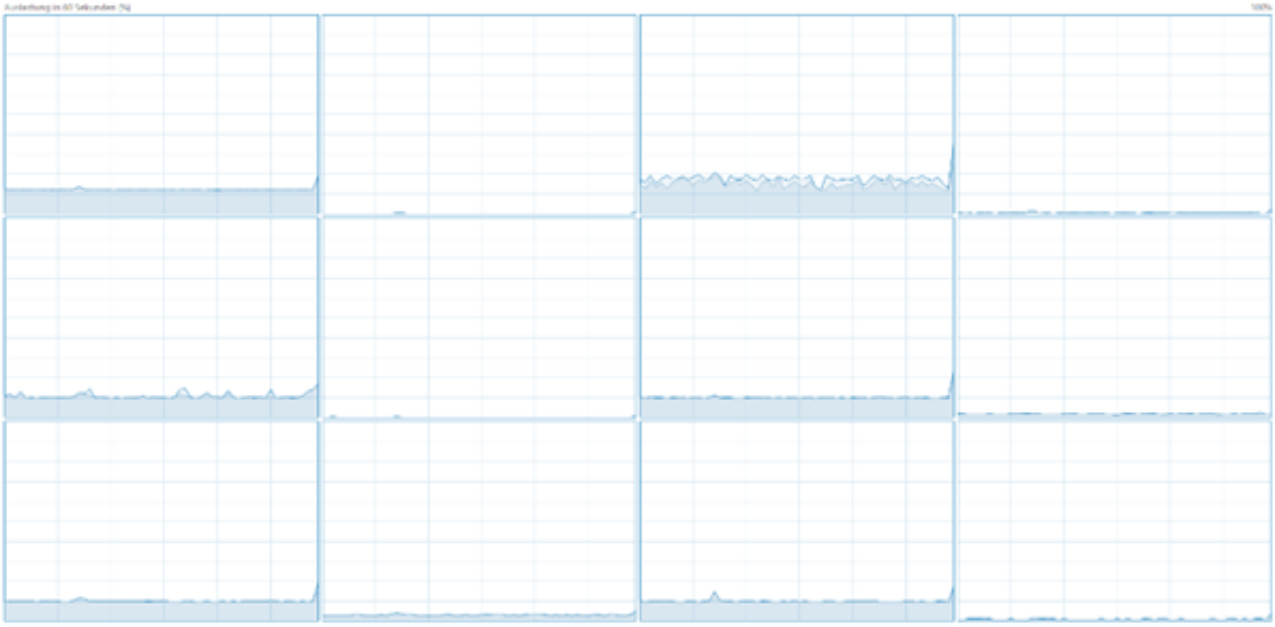
In order to counteract this problem, MATRIX VISION developed a software solution which gives the user greater control over the work distribution to the CPU cores.

Preventing data loss by using the Multi-Core Acquisition Optimizer (MAO)

With the Multi-Core Acquisition Optimizer (MAO) the CPU cores can be optimally utilized and a possible data loss can be prevented.

With the Multi-Core Acquisition Optimizer the distribution of the workload arising from the processing of the network data can be influenced. It allows the processing of the network data of a GigE Vision Stream to be distributed to several CPU cores through the intelligent combination of device firmware and host driver. In contrast to processing on a pseudo-random selected CPU core, with the Multi-Core Acquisition Optimizer it can be explicitly defined on how many and which CPU cores the processing of the network data takes place – according to the

individual requirements of the respective application. The distributed workload ensures enhanced overall system stability and data loss during transmission is prevented.



3rd caption: Network data processing of a camera distributed to several CPU cores.

The Multi-Core Acquisition Optimizer not only leads to more stability, but also increases the processing power. The distributed workload of the data processing ensures that the work packages of the parallel algorithm can be bigger and thus free capacities of the CPU cores can be better utilized.

4th caption: The entire system performance can be utilized thanks to the distribution of the network data processing

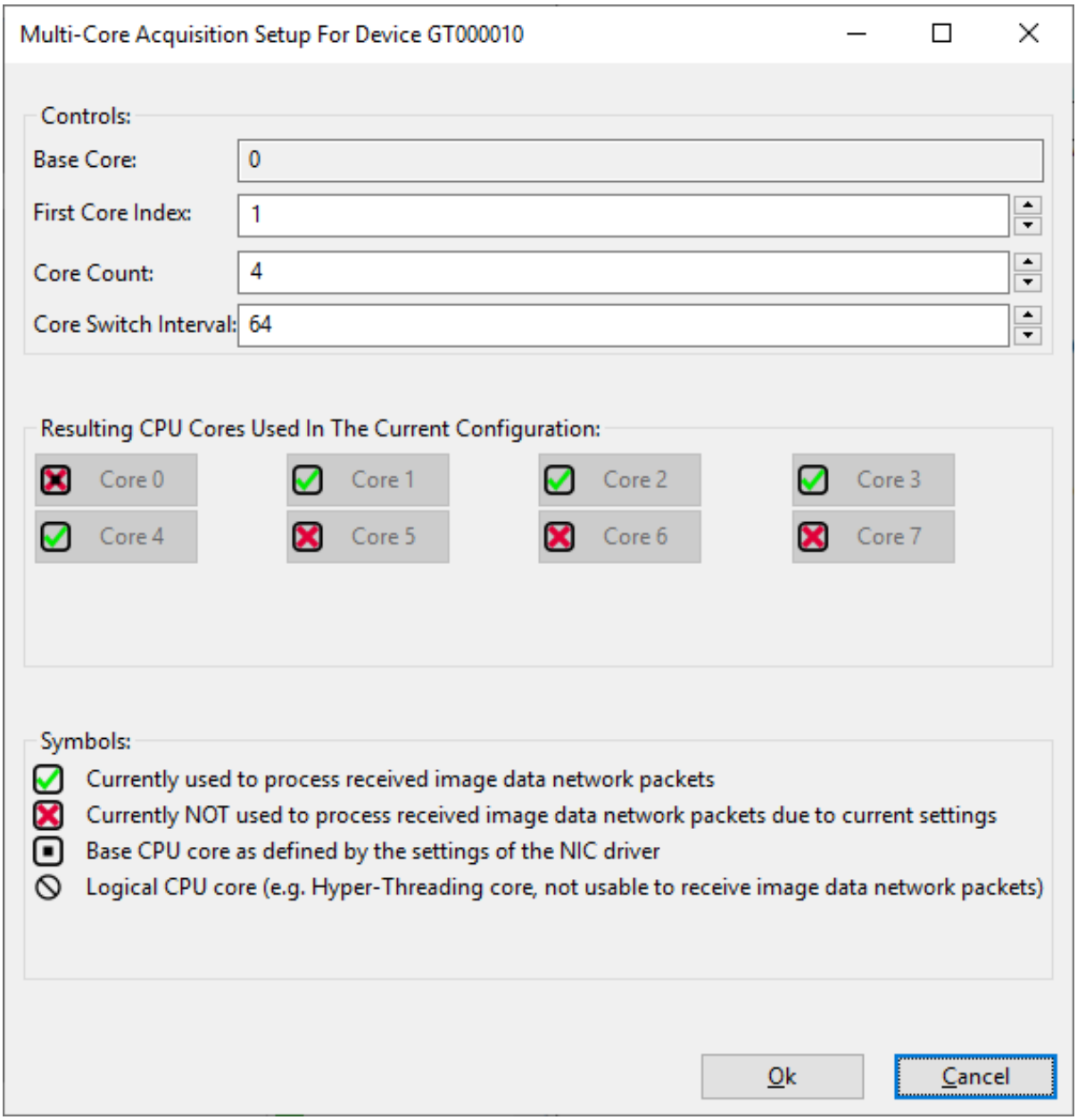
The distribution is realized in such a way that the host system can deal most effectively with it. With the knowledge of on how many and which CPU cores the processing of the network data takes place, it is also possible to unlink this processing from the rest of the machine vision application. This is achieved by omitting the CPU cores dedicated for the processing of network data from the processing load by the parallel running machine vision application.

MATRIX VISION has solved this problem on the software side, which was previously only feasible with an expensive hardware upgrade. Without any additional expense and effort, you can use your 10GigE camera from MATRIX VISION at maximum frame rate without any concerns and run a time-consuming algorithm for processing the images at the same time.

The Multi-Core Acquisition Optimizer is available via the MATRIX VISION device firmware in combination with the current mvIMPACT Acquire driver version. There are no additional costs for customers. It is configured and activated via the GUI tool wxPropView and the relevant wizard or the mvIMPACT Acquire SDK.

Procedures

The wizard itself is rather simple:



- **"Base Core"**: This is a **read-only** feature! It reflects the RSS base processor number as configured by the user (if supported by the NIC driver) or as selected by the NIC driver (if either **NOT** selected by the user **OR** if manual selection is not supported by the NIC driver). CPU cores smaller than this value cannot be used for processing network data from devices connected to this NIC. More details about RSS and this parameter in particular can be found in various places on the Internet.
- **"First Core Index"**: This parameter allows the selection of the first CPU core that shall be used for processing network data from this device relative to the **"Base Core"**. E.g. if the base core has been set to 3 then setting the first core index to 2 will result in CPU core 5 (3 + 2) being the first core to use for data processing.
- **"Core Count"**: This parameter defines how many CPU cores starting at **"First Core Index"** shall be used for processing image network data coming from this specific device. Using more cores reduces the overall load

for each individual core but also results in a slight overhead caused by core switches that must be handled by the system

- **"Core Switch Interval"**: This parameter defines after how many network packets another CPU core shall be used. This value is only important when using more than 1 CPU core for processing network data! Selecting a value that is too high can result in data loss as an individual core might not be able to get rid of its received data in time and selecting a value too low introduces too much overhead for core switches.

As a general rule of thumb the following guidelines can be used:

1. If a single CPU core can cope with all the data the best performance can be achieved by using a dedicated CPU core for processing the network data and by removing all other application specific CPU load from this core. So explicit selection of 1 core is done using the Multi-Core Acquisition Optimizer feature (without it the CPU core used by the system will be different for each session) and the application itself uses appropriate mechanisms to move the remaining load to the other cores (e.g. by setting the process affinity mask)
2. If the application cannot explicitly move its work to certain cores or when a single core is not powerful enough use 2-8 cores in parallel for network data processing and select a higher **"Core Switch Interval"** when using less CPU cores

When the wizard is configured its effect can e.g. be observed using the Windows task manager: