The aim of the project is the development of computation organization in reconfigurable Multiprocessor System on Chip (MPSoC), which will enable the realization of real-time processing of UHD/4K video stream (3840 x 2160 pixels @ 24/30/50 frames per second).

Nowadays a very dynamic development of cameras can be observed. Devices which support 1920 x 1080 pixels resolutions (so-called Full HD) are widespread. What is more, recording with 3840 x 2160 pixels resolutions (i.e. UHD/4K in 16: 9 format) is available in the last generation of popular sport cameras and camcorders. Also, today's smart phones are equipped with a 12 and more megapixels sensor, which allows to capture 4K videos. Additionally customer-grade displaying devices, like 4K monitors or TVs are already available on the market for some time.

While the acquisition and display of 4K video stream in now fully supported, performing image processing operations with the commonly used computing platforms (i.e. personal computers and notebooks) is very time consuming and requires a lot of disk memory. One can be convinced of this, when editing this type of video. One should realize that 1 uncompressed colour 4K frame equals about 23 MB of data and a 1 minute video (30 frames per second) more than 41 GB. However, in many cases the use of high resolution video stream could improve the performance of the considered vision systems. A good example is pedestrian detection used in recently very popular advanced driver assistance systems (ADAS). On a high resolution image we are able to detect persons, who are quite far away from the camera and this enables to detect a potentially hazardous situation much earlier.

At this point, we could ask the following question – is it possible to realise real-time processing, e.g. pedestrian detection for 4K video stream with 24/30/50 frames per second. It should be also noted, that in the considered example, as well as many others, such as drones (UAVs), electric vehicles or video surveillance, we cannot use powerful computing platforms like supercomputers or GPUs because they consume too much energy. A very promising solution seem the latest generation of reprogrammable devices (FPGA) additionally equipped with a powerful processor system and graphic processor (e.g. the Zynq Ultrascale+ family by Xilinx). They allow to perform parallel data processing, which is essential for obtaining real-time performance. Additionally they can be used very close to the vision sensor (constituting the so-called smart camera). Therefore, the huge data stream does not need to be transmitted, as all required processing is performed locally. Moreover, a distributed computing system is also more energy efficient.

During the project inter alia the following elementary vision algorithms will be considered for real-time implementation: image filtration (averaging, Gaussian, median, but also more advanced methods like vector median filter, bilateral filter), edge detection (Sobel, Canny), single-pass and double-pass connected component labeling, as well as depth map estimation (stereo-vision, image rectification, structure from motion), moving objects segmentation (optical flow), foreground object segmentation, object detection (e.g. people, cars) and object tracking. With these components, it is possible to build advanced vision systems running up to 4K resolution e.g. abandoned luggage detection or re-identification of people in a multi-camera surveillance system.

Moreover, a methodology for transforming a high level language (C/C++) description of a vision algorithm into an efficient hardware-software implementation in a programmable MPSoC devices will be developed. In addition, the proposed approach will be compared with tools, which perform this computing resource assignment automatically. This experiments should provide an answer to the question, whether and for which algorithms it is possible to obtain a hardware-software computing system capable of processing 4K images in real time.

In this project video stream processing is considered, as it is a very nice example to show all the advantages of hardware-software solutions. However, the results, methodology and know-how could be also used in other applications that require fast data processing (e.g. biomedical), especially when embedded systems are considered.