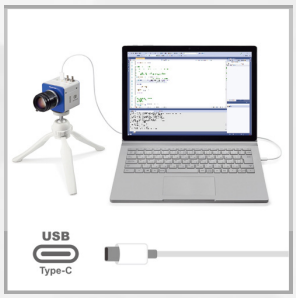
Photron

INFINICAM

INFINICAM is a high-speed streaming camera capable of transferring 1.2-megapixel image data to PC memory at 1,000fps via USB 3.1. Image data transferred to PC memory can be processed to easily build a high-speed imaging processing system. The latest version of the SDK supports Python, a common language in the computer vision/machine vision field, in addition to C++. This makes real-time image processing with INFINICAM possible with easier and more intuitive programming.

Product Features

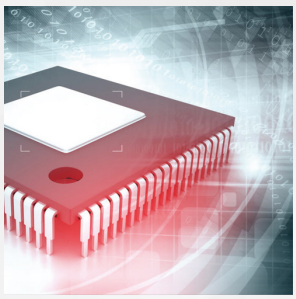


High-speed image processing made simple

High-speed image processing used to require a dedicated board and complex cabling, but now an environment can be created with a single general-purpose USB 3.1 Type C cable and a single PC.

Compact, Lightweight, C-mount

The compact and lightweight housing, (55 mm x 55 mm x 55mm, 280g), makes it easy to install. The C-mount lens compatibility allows for a wide range of lenses to be selected.



Real-time high-speed image compression

Real-time compression to less than 1/4 the data volume enables high-speed image transfer at 1,000fps and up to 30,000fps at a resolution of 1,246 x 1,008 pixels, which cannot normally be sent using USB 3.1.

Open development environment

The latest SDKs, development environment, and manuals can be downloaded from the web. The source code of sample programs will also be made available in due course.

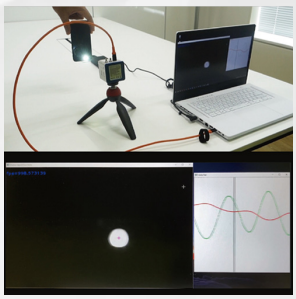
python™

Example: Initializing the camera and capturing a single image.

```
cam = CameraFactory ().create()
decoder = cam.decoder()
xferData = cam.grab()
img = decoder.decode(xferData)
```

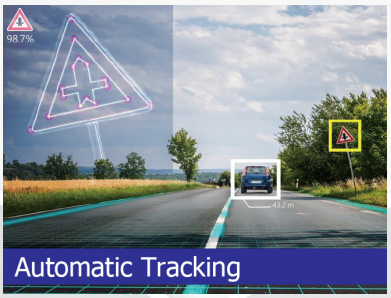
Data capture with just four lines of programming

Python is now supported, making programming easier and more intuitive. For example, image acquisition can be coded, as shown in the image to the left, for quick implementation.



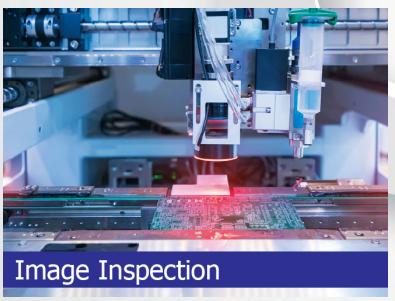
1,000fps real-time image processing

A published sample application can be seen in the example to the left. Here you can see the capture of the LED lights of a smart-phone and calculated output, along with XY coordinate values in real-time. The sample app comes with source code (C++ and Python.)



Object Detection
Object Tracking
Motion Analysis
Object Labeling
Template Matching

Focus Detection
Edge Detection
Surface Roughness Analysis
Optical Flow
Digital Holography



Product Specifications

Model Name	INFINICAM UC-1
Sensor Type	CMOS Image Sensor
Sensor Size	12.8mm x 10.24mm
Pixel Pitch * Square Pixels	10µm
Maximum Effective Resolution	1246 x 1024
Maximum Shooting Speed (Full Frame)	988fps
Maximum Shooting Speed (Split Frame)	31,157fps
Minimum Exposure Time	6.5µsec
Shutter Method	Global Shutter
Density Gradation	Monochrome 8-bit
Interface	USB 3.1 Gen 1 Type-C
Lens Mount	C-mount
External Synchronisation Signal	2.5 Vp-p (DIN connector male)
Camera Housing	Unsealed air-cooled (with fan)
Dimensions / Weights	55(W) x 55(H) x 55(D) mm / 280g (excluding protruding parts and accessories)
Storage Temperature / Humidity	20 to 60°C/85% or less (no condensation)
Operating Temperature / Humidity	0-45°C/ 80% or less
DC Power Supply	5V (USB Vbus supply)

Development Environment

OS	Microsoft Windows 10 64-bit
CPU	AVX2 - compatible processors
Runtime	Visual C++ 2019 Redistribution Package
Supported Programming Languages	C++, Python

Imaging Performance

Resolution	Frames Per Second (fps)
1246 x 1024	50
1246 x 1024	250
1246 x 1024	500
1246 x 1024	988
1246 x 1008	1,000
1246 x 496	2,000
1246 x 176	5,000
1246 x 80	10,000
1246 x 32	20,000
1246 x 16	31,157

* Shooting speed can be set from 1 to 31,157fps.
* The exposure time can be set in 0.01µsec increments to 6.5µsec.
* The resolution can be set in increments of 16 pixels height. (Limited for split frames).
* The set shooting performance may not be achieved depending on the development environment.

Main Functions of the SDK

- Initializing the library
- Searching for devices
- Open and close devices
- Acquisition and setup of shooting speed/shutter speed
- Clock settings for exposure/non-exposure times
- Acquisition and setting of synchronous signal inputs
- Start and end of continuous transfer
- Obtaining and setting the number of ring buffers
- Image acquisition
- Obtaining and setting quantization tables
- Decoding of compressed data

Dynamic

Exclusive Authorized South East Asia Distributor

Dynamic Analysis System Pte Ltd
www.photonics.com.sg
Email: sales@photonics.com.sg