

3D Cameras Benchmark for Human Tracking in Hybrid Distributed Smart Camera Networks

Andres Burbano^{#++}, Marius Vasiliu^{#+}, Samir Bouaziz^{#+}

[#]SATIE - CNRS UMR 8029

⁺Paris-Sud University, Paris-Saclay University, France

^{*}R&D, Shopline Electronic, 78220, Viroflay, France

{andres-alberto.burbano-lopez, samir.bouaziz, marius.vasiliu}@u-psud.fr

ABSTRACT

This article presents a comparison between the different types of depth sensors that are currently on the market after the unavailability of the *PrimeSense* technology. Several academic research projects in 3D computer vision use this sensor technology. In order to find a substitution to this camera in terms of quality and price, we developed a study of the available 3D geometric sensors that can be used in real-time and low-cost applications. This work is part of our SHuKB SDK for detecting and counting people in hybrid smart camera networks. Our main contribution pertains the quantitative and qualitative assessment of the accessible sensors on the market to improve products identification, close to our needs and provide feasible replacements. This objective intends to avoid the consequences of a sudden loss of sensor products, such as the *PrimeSense* sensor. This assessment is based on a set of properties and relevant features for people detection and sensing. We present the results of this study and provide an analysis of the different 3D sensors, along with our vision of their fields of application, focused on the smart camera networks and people tracking.

CCS Concepts

•Computing methodologies → Computer graphics; Image processing; •Computer systems organization → Embedded systems; •Hardware → Coverage metrics; Equivalence checking;

Keywords

3D Sensing; Depth images; Stereo camera; Human Detection; Smart Camera Network

1. INTRODUCTION

Smart cities, smart buildings and smart environments often use images combined with additional sensors. The image sensor is the main component of the so-called smart cameras [20]. The amount of different technologies and the

miniaturization level allowed constructors to develop several types of cameras, depending on usage constraints, merging techniques and providing several solutions for image sensing. Simultaneously, the maturity of these technologies gave way to a new generation of 3D sensors. Following the appearance of the Kinect, a low cost depth sensor with satisfactory accuracy, an incredible amount of depth-based researches were conducted in most major research fields of image processing, including smart environments [7, 3]. The success of this 3D camera concept incremented research and evolution, attracting several global manufactures to build their own solutions. Due to insufficient available comparative studies between manufacturers and cameras performance information, we decided to quantitatively and qualitatively assess the current 3D detection cameras on the market. This paper analyses comparisons of different cameras and their characterization in order to evaluate the performance, reliability and possible applications usage. Our goal is to encourage cameras' early characterization, to identify their suitable applications, address cross-cutting issues and accelerate depth sensing research.

Simultaneously, this evaluation addressed the demand in our work [4], by identifying cameras that are capable of acquiring images in indoor and outdoor environments. This work describes a 3D-Sensing distributed embedded system for people tracking. Additionally, it defines a scalable and dynamic software architecture based on a component bricks assembly. These bricks are interchangeable and accomplish specific tasks of the system. Moreover, this architecture allows to build bricks in order to overcome technology limitations or to standardize the system's processes.

Our main contributions are the cameras characterization methodology, the analysis and the experiment on different recently released low-cost 3D sensing cameras and their evaluation. Furthermore, we propose an expansion for the SDK described in [4], by implementing a camera I/O brick and a *fast camera characterization toolkit* (FCCT) without our methodology. The cameras characterization is done by a comparison (giving an overview of the technical specifications and technology development environment) and an evaluation using three different procedures focused on the smart camera networks and people tracking applications. These experiments assess the accuracy and precision of camera measurements, the signal stability and the humans detection reliability. The I/O brick allows the integration of new cameras to the hybrid smart camera networks efficiently, and grants access to the system's functionality. As it was previously stated, the FCCT provides a series of software tools

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ICDSC '16, September 12 - 15, 2016, Paris, France

© 2016 Copyright held by the owner/author(s). Publication rights licensed to ACM. ISBN 978-1-4503-4786-0/16/09...\$15.00

DOI: <http://dx.doi.org/10.1145/2967413.2967431>

to facilitate the cameras characterization.

This paper is organized as follows: Section 2 describes related works. Section 3 describes and compares the cameras technical characteristics and functionality. Section 4 describes the evaluation process: experiment procedures and results. Finally, Section 5 presents the conclusion of our work.

2. RELATED WORK

Reisslein et al [18] describe *smart camera network* as an assembly of "computer vision, image sensors, embedded computing, and sensor networks" technologies. The main challenges in their design are: *system design (hardware and software)*, *video analysis*, *sensing*, *privacy-protection*, *hybrid sensor networks*, *network deployment and higher-order data processing and management*, compiled from [18, 19, 20, 4].

Lately, several smart cameras tend to use a 3D sensor as the *image sensors* solution. Additionally, in recent years, considerable works using depth sensors technology [5, 11, 14] were developed, including our previous work [4] that describes a smart camera network where each node is composed of a depth sensor, an embedded computer and a communication unit, to build camera networks. Therefore, we considered it useful to create a depth camera characterization methodology. Previous works related to cameras evaluation are found in the literature. The paper [9] describes the technology of the Time-of-Flight (ToF) of different 3D cameras. In [17] we found the comparison between both Microsoft sensors generations. In [22] the authors describe the signal stabilization of 3D position measurement. Furthermore, Nair et al.[16] proposed a ground truth datasets characterization, identifying an important set of requirements focused on the algorithms such as low latency, low noise or depth range. However, despite the growing interest in depth sensing technology and applications, a general evaluation method focused on the depth image sensor evaluation has not been proposed so far. Therefore, we planned to create an evaluation methodology and implement a toolkit capable of performing the procedures explained in section 4.

3. DEPTH CAMERAS

This section describes and compares several cameras selected from the market. The selection criteria considered in depth, real-time image acquisition facility, low-cost, recent release and feasibility of integration into a smart camera network node and usability in different environments.

The cameras' description contains *signaling* [21], the technology description, patent or reference, the amount and type of sensors, the framework or SDK additional functionality and the power supply source when it is different from the USB cable connection.

3.1 ASUS Xtion Pro

The Asus Xtion Pro is an active camera, based on structured light pattern, described in [9]. This camera is composed of a fixed pattern IR emitter, an IR camera and an auxiliary RGB camera. It can acquire depth and color images at the same time. The provided SDK is the well know OpenNI that allows controlling the Kinect and the PrimeSense Carmine (all three cameras use the same sensor). This camera can only work in indoor environments and it has been integrated to a *smart camera* system [4]. Additionally,

we selected this camera to construct our **frame of reference**, due to the vast literature based on this RGBD sensor, including the compiled in [7].

3.2 Microsoft Kinect for Xbox One

This camera, also known as the Kinect 2.0, is an active camera based on time-of-flight technology[17]. It is composed of an infrared (IR) emitter, an IR camera and an auxiliary color (RGB) camera. It has the Kinect for Windows software development kit (SDK) 2.0 that includes examples and access to the source code. Additionally, this SDK supports .NET languages. This module is powered by a proprietary Microsoft power supply.

3.3 Intel SR300

The Intel SR300 is an active camera based on Coded Light [2]. It is comprised of one IR camera, an IR pattern projector and an auxiliary 1080p RGB rolling shutter sensor. It has a complete RealSense SDK for Windows and a cross-platform API called libRealsense. The API allows changing sensors properties and stereo algorithm parameters. It is possible to configure the following aspects of color and infrared cameras: the brightness, gain, exposure, frames per second (fps), white balance, auto-exposure and auto-gain. Furthermore, it is possible to set up a region of interest in the image in order to calculate automatic values. Additionally, the Real Sense Driver provides compatibility to the Robotic Operating System (ROS) and it provides a complete set of examples source code.

3.4 Intel R200

The camera Intel R200 is an active/passive stereo camera based in structured stereo [1, 10]. This camera is composed of two IR cameras and an infrared laser that projects texture, improving short distance measurements in indoor environments. In addition, it provides a 1080p rolling shutter RGB camera, and it uses the same SDK as the SR300 model. This module is powered by USB, but is restricted to short distance cables due to power consumption. Sadly, the Intel cameras only work with Intel Processors.

3.5 University of Bologna Camera

The University of Bologna solution is a passive stereo camera with 2 RGB sensors and an FPGA that calculates the *disparity map*. This camera can be used in indoor and outdoor environments and has already been integrated into a *smart camera* implementation [14]. Further information can be found in [15]. The camera's design allows modification of the baseline distance and imaging sensors between RGB and grayscale. It has a proprietary library that provides the image acquisition functionality and parameters setup.

3.6 Etron eSP870

Etron eSP870 chipset is a USB 3.0 and 2.0 depth-map controller, ready to build stereo cameras, which delivers color and depth images [12]. Etron provides an eSPDI library for Linux, embedded Linux and a more complete framework for Windows. The Windows framework provides post-processing functionality, performed by the host CPU. The used model is composed of two HD 720P (1280x720) CMOS image sensors and the eSP870 chipset. The controller provides a pair of synchronized 1080p color images at 60 fps using USB 3.0 and delivers VGA depth images up to 30 fps

in Windows and up to 15 fps in Linux. However, during our test the camera was not able to provide a 1080p pair of images and the depth images simultaneously. Currently, Etron manufactures 3 cm and 12 cm baseline camera models.

3.7 Parrot Kalamos

The recently released navigation solution developed by Parrot is known by the code name Kalamos. The 3D sensor system is composed of two high-definition fisheye cameras. Each camera has a 1500x1500 pixels resolution. The disparity map is calculated with the central pixels of the images at 640x480 pixels resolution. This solution provides compatibility with the Robotic Operating System (ROS). It can be used as a sensor or an autonomous navigation system, thanks to the inertial measurement unit (IMU), ultrasound sensors, a Tegra K1 (NVidia) processor and an embedded Ubuntu OS. Due to the camera's nature, it is suitable for indoor and outdoor environments. The camera is powered with a 12V power supply.

3.8 Sony PS4eye

The PS4eye is a synchronized pair of color CMOS sensors. It is a proprietary solution for the Sony PS4 game console. However, an open source driver was built with reverse engineering and is available on the web. This camera is powered by the AUX port, a proprietary connector from Sony. In order to test the camera, the cable had to be modified for a standard USB 3.0 interface. This camera only provides color synchronized images. The disparity maps have to be calculated by the host CPU.

3.9 Stereolabs ZED

This high-resolution depth sensor proposed by Stereolabs provides a pair of color cameras with a maximum resolution of 2208x1242 pixels resolution at 15 fps and up 100 fps with a VGA resolution. The ZED SDK processes the disparity map on the host machine requiring a recent NVidia GPU with CUDA API 6.0. Other host minimum requirements are a dual core CPU 2.4 GHz and 4GB RAM. The camera was designed mainly for autonomous navigation and mapping.

3.10 Cameras Comparison

The cameras comparison is showed as follows: Table 1 lists the operating systems supported by each camera's framework, Table 2 presents the languages supported by the camera SDK or API and Table 3 compares the camera's general specifications.

This work aims to find suitable cameras to integrate into hybrid smart camera networks. Therefore, the SR300, the ZED camera and Kinect 2 were not included in the following evaluation section as the PS4eye has to be physically modified to change the AUX connector (proprietary) to a USB3 connector to acquire images. In addition, the available driver was developed using reverse engineering which consequently does not provide any depth image directly from the sensor. Therefore, the disparity calculation has to be done by the host CPU as the ZED camera. This requirement definitely restricts its use in low-cost embedded solutions like the smart camera networks. However, the ZED camera has a complete official SDK working with resolutions up to 2208x1242 pixels. On the other hand, the Kinect's main drawback is the proprietary power supply and energy consumption requirements to integrate into an embedded

Table 1: Supported OS

Camera SDK/API	Embedded Linux	Linux	Windows	Mac OS
Xtion Pro	x	x	x	x
Kinect v2	x	x		
SR300	x	x	x	x
R200	x	x	x	x
UniBo	x	x	x	
eSP870	x	x	x	x
Parrot	x	x	x	x
PS4eye	x	x		x
ZED	x	x	x	

Table 2: Programming languages

Camera SDK/API	C	C++	Java	Other(s)
Xtion Pro	x	x	x	x
Kinect v2		x		.NET
SR300		x*		
R200		x*		
UniBo		x		
eSP870	x	x		
Parrot		x		Python
PS4eye		x		Python
ZED		x		

* Requires a C++11/14 compiler.

device. We consider its size as a disadvantage when using it in the smart cameras field. Nevertheless, a detailed evaluation of the Kinect 2 can be found in [17]. Regarding the SR300, its range of view (Table 3) is too short and the Coded Light pattern generates destructive noise, preventing its integration into sensor networks. The camera developed by the University of Bologna was not available prior to the deadline.

4. CAMERA CHARACTERIZATION

In this section, we describe a set of procedures for depth cameras characterization. We expand on the cameras evaluation in [17] and proposed 3 main experiments. First, we analyzed the depth signal stability over the time. Then, we studied the precision and the accuracy of depth measures of the camera with respect to the ground truth. Finally, we estimated the amount of pixels covering a human person rate depending the position in the field of view, when a camera is oriented in a downward position. This last point specifically addresses our goal to perform people detection and counting in public space.

The cameras' tested resolution was 640x480 pixels and the frames per second acquisition performance was of 30 fps, except for the Etron camera, which had a maximum performance in Linux of 15 fps. The camera exposure and automatic gain were stabilized before the depth frames acquisition.

The computer hardware used in our experiments presents an Intel Core i7 4712MQ CPU at 2.3GHz, 16GB of DDR3-RAM, and a 250GB solid-state drive and a Linux Ubuntu 14.04 distribution OS. The processes were carried out in an indoor environment, in an enclosed space. We assumed as in [17] that no major environmental changes would occur during the depth data acquisition (with a typical duration of less than one minute).

4.1 Temporal Depth Stability

Table 3: General depth cameras specifications

Specifications	Xtion	Kinect v2	SR300	UniBo	R200	eSP870	Parrot	PS4eye	ZED
Depth Sensors	IR	IR	IR	2 RGB	2 IR	2 RGB	2 RGB	2 RGB	2 RGB
Resolution	VGA(D)	512x424D	VGA	VGA	VGA	HD	1500x1500	WXGA	2208x1242
Depth Shutter	Global	Global	Global	Global	Global	Global	Rolling	Rolling	Rolling
Auxiliary Cameras	RGB	RGB	RGB	N/A	RGB	N/A	N/A	N/A	N/A
Resolution	VGA	FHD	VGA		FHD				
Color Shutter	Global	Global	Rolling	N/A	Rolling	N/A	N/A	N/A	N/A
Depth Measure	Active	Active	Active	Passive	Mix	Passive	Passive	Passive	Passive
Calibration	Prop.	Prop.	Prop.	OpenCV	Prop.	OpenCV	Prop.	Prop.	Prop.
FoV WxH (deg)	58x45	70x60	68x54	var.	59x46	62x38	200x200	85x53	96x54
USB Interface	2	3	3	2/3	3	2/3	3	Aux/3	2/3
Baseline (mm)	74	25	47	var.	70	120	200	85	120
Indoor range(m)	0.8-4	0.5-4.5	0.20-1.50	1-5	0.7-3.5	1-10	1-5	0.3-inf	1-15
Outdoor range(m)	N/A	N/A	N/A	1-5	1-10	1-10	1-5	N/A	1-15
Frame rate(fps)	30	30	30	22	30	15	30	7.5-240	15-100

Due to the nature of the depth data computing [9, 10, 13], a fixed camera looking at a static scene produced a noisy depth signal between consecutive frames [17]. The objective of this experiment is to estimate the depth maps variation, or in other words, depth map signal stability.

4.1.1 Procedure Description

In the first procedure we recreated a suitable scene where most of the surfaces were parallel to the camera’s plane, avoiding the occurrence of *flying pixels* [9] due to the scene in-homogeneity between the background and the objects. In addition, we covered the non reflecting surfaces with a suitable reflective material to isolate the technology issues from the algorithm evaluation. We measured the *root mean square error RMSE* of the depth signal through time using T consecutive frames D^t , $t = 1, T$. For each pixel p belonging to the set R of valid depth pixels, let $D^t(p)$ be its depth value in the frame t and let $\bar{D}(p)$ be its temporal mean through the whole frame sequence. Then:

$$RMSE(p) = \sqrt{\frac{1}{T} \sum_{t=1}^T (D^t(p) - \bar{D}(p))^2} \quad (1)$$

In order to have a global camera behavior indicator, we defined the *Global Signal Stability* as:

$$GSS = \frac{1}{|R|} \sum_{p \in R} RMSE(p) \quad (2)$$

where $|R|$ is the cardinal number of set R . These global indicators (presented in Table 4) characterize only the temporal noise of the depth signal and not its accuracy.

4.1.2 Test Results

Figure 1 presents the RMSE of 1800 consecutive depth frames and its histogram for each evaluated camera. A lower RMSE denotes a more stable depth signal. All cameras are directed to the static scene from the same point. This scene was composed by two big plane square surfaces parallel to the camera image plane and one plain background. The following analysis considered the differences between the cameras fields of view, the temporal measurement stability and a number of pixels acquired per camera.

The Xtion Pro Camera FOV was our reference. In the resulting image (Figure 1a), we found the majority of the measurement variation in the contour area of the objects, while

Table 4: Global results for depth sequences

Pixels	Valid %	Outliers %	Invalid %	GSS(mm)
Xtion Pro	98.3	0.5	1.2	0.77
R200	66.6	13.2	20.1	14.53
Etron	67.4	17.3	15.3	6.19
Parrot	90.9	5.7	3.4	20.64

the plane areas were significantly stable. We also found some measurement variations on frame border. The majority of the pixels of the image contained valid depth information.

The R200 camera (see Figure 1b) has the same FOV as the reference. The image showed unstable measurements even in the plain surfaces of the closer objects, discarding a flying pixels problem. The instability of the R200 responds to the nature of the depth map estimation process, which depends on the projected texture and the algorithm parameters setup. The closer measurements are more stable, while further object measurements become noisier until the point to get invalid pixels represented in black (no disparity calculated).

The Parrot (see Figure 1c) camera sees more than double the vertical space and a wider horizontal area with respect to the reference frame. The resulting disparity map is composed of 10x10 pixels tiles (constant depth in each tile). The closer objects had a stable behavior, while the rest of the image had deformation and instability problems. As a result of the fisheye border effect, several pixels from the image had no information.

The Etron camera (see Figure 1d) has the smallest FOV. The measurements in the plain surfaces are more stable than the Parrot and the R200 camera. We found the main measurement errors on contours and at furthest distances (top part of the image). Additionally, many pixels have no depth data on a narrow vertical band on the right side of the depth map.

In general, the major measurement instability was found at the border of the objects due to occlusion and flying pixels. The stereo cameras had less information as a consequence of the technology’s nature. Moreover, the Etron camera measurements were found to be more stable than their peers (stereo-based cameras) despite the smaller FOV. The R200 has a larger FOV than the Etron, yet the R200 signal stability decreases the further the distance. Finally, the wider FOV of the Parrot camera allowed to track larger areas but the tiled depth map prevents a detailed human detection.

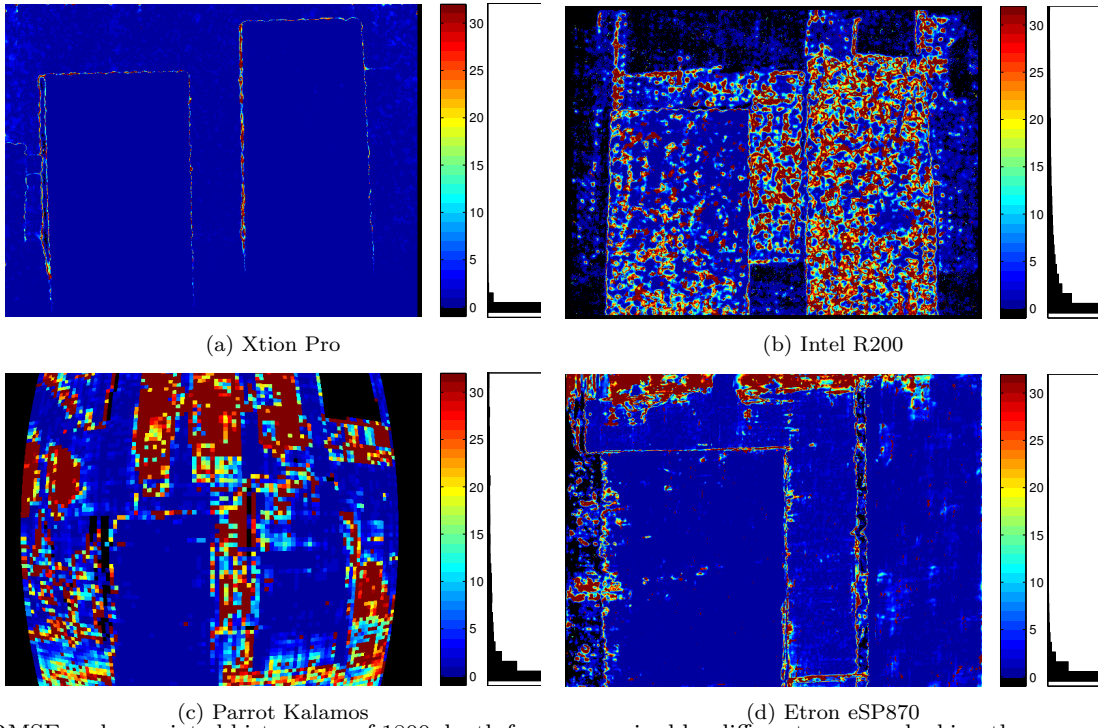


Figure 1: RMSE and associated histograms of 1800 depth frames acquired by different cameras looking the same static scene. The scene is composed by two flat rectangular surfaces at 2 and 4 meters away from the camera. Each RMSE image uses the same color-map in the range 0 .. 30 mm. The outlier pixels (with a depth error higher than 30 mm) are in dark red and the invalid pixels (with no depth data from the sensor) are in black.

4.2 Depth Map Precision and Accuracy

Due to the depth map calculation algorithms, the disparity uncertainty increases at further distances from the camera [8]. Consequently, the error between the measured depth and the ground truth increases at longer distances. Conversely, robust algorithms tend to calculate regular depth maps in steady scenes for every frame. Therefore, we carried out this experiment to evaluate the camera’s accuracy, precision and nominal range.

4.2.1 Procedure Description

In the second experiment we shifted the 3D sensor along an axis orthogonal to a reference wall, keeping the camera’s optical axis orthogonal to this wall. The typical shift step was equal to 40 cm, between 80 cm to 520 cm from the wall. For each step s we acquired T depth frames from a flat surface in order to avoid *flying pixels*. At the same time, we ensured that the measured pixels were at the same distance to the sensor, the camera and wall planes were parallel. For each step s , we estimate the *accuracy error* E_s of the depth data for a specific region R^c as the difference between the mean depth of this region $\bar{D}_s$ through time and a manually measured distance d_s :

$$E_s = \left[\frac{1}{|R^c|} \sum_{p \in R^c} \bar{D}_s(p) \right] - d_s \quad (3)$$

In order to evaluate the optical distortion influence to the accuracy error, we consider for R^c two different specific regions in the image: one at the optical center and the other close to border. We proceeded by evaluating equation 3 at each step, for both regions and for each camera.

4.2.2 Tests Results

The above described procedure allowed us to estimate the camera’s accuracy over the distance, distortion influence and the nominal range (minimum and maximum distance where camera measurement is reliable).

In Figure 2, we presented the camera’s precision. In Figure 3 we show the comparison of the cameras accuracy over distance. This figure shows the cameras accuracy behavior over the distance of each camera. The Y axis represents the E_s and the X axis represents the measured distance d_s . This figure shows the cameras’ nominal range.

4.3 Human Detection Reliability

4.3.1 Procedure Description

The third experiment evaluates the human detection reliability HDR and the perspective influence of 3D-camera oriented in downward direction. The HDR is measured by the amount of depth pixels acquired and associated to a human person in the field of view, compared to the real amount of pixels that the person occupies in the image (using RGB or auxiliary cameras of the 3D-sensor). We chose to put the camera in the downward position in this test in order to minimize occlusion, to optimize the coverage and to increase the amount of points acquired per person[4][6]. Simultaneously, we filtered the pixels under the chest high of the person. The perspective influence is measured by analyzing the size of the segmented region for a human person at different positions in the FOV and looking at different directions. In this procedure we divided the FOV in three equal regions R_n horizontally like in Figure 4a. We define 4 different directions ϕ_m described in Figure 4b.

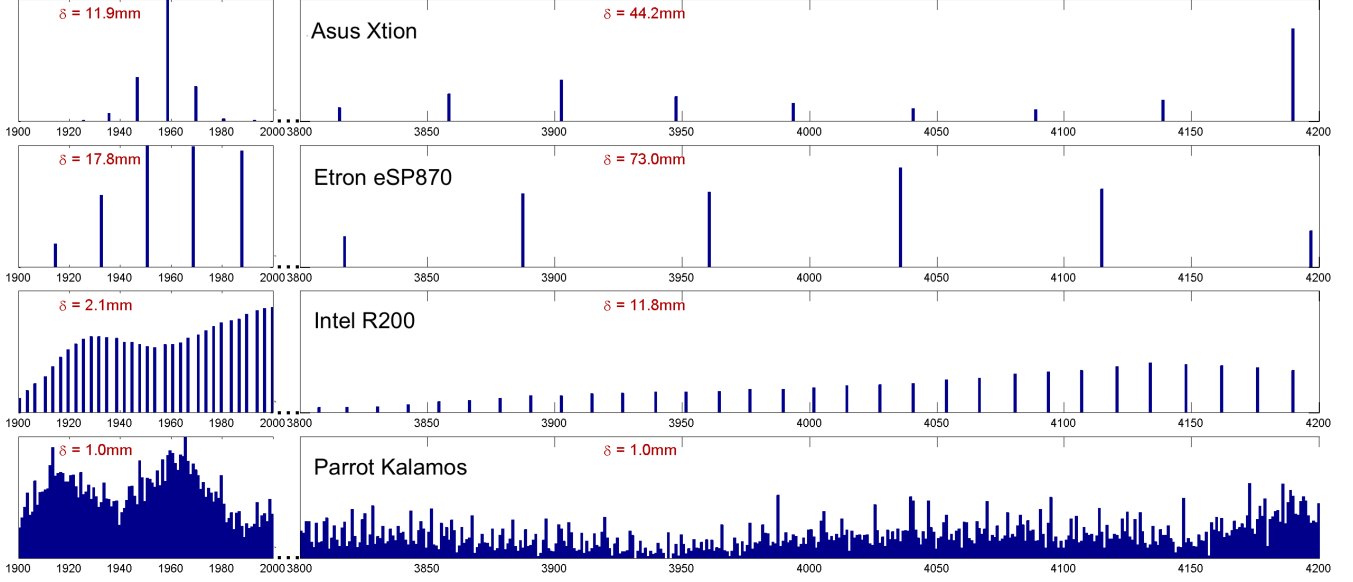


Figure 2: Depth data histogram for several cameras. Zoom windows at ranges 1900-2000 mm and 3800-4200 mm show the local precision of depth for each camera technology. Vertical axis uses a normalized scale, meaningless here.

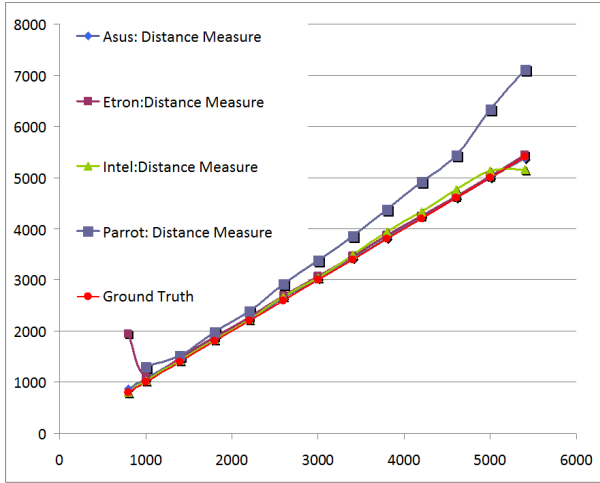


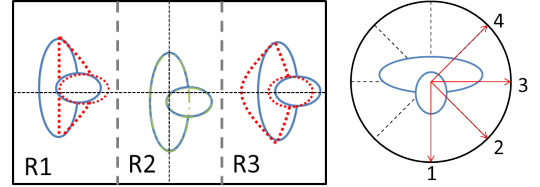
Figure 3: Camera Depth Accuracy

Each sample S_m^n consisted of placing a person in each R_n looking at each ϕ_m direction. At each position, a synchronized color and depth image were acquired. We ensured that every S_m^n images contained the person and the background. We used an easy-to-identify color for the background to facilitate the person segmentation in the color image. Then, we counted the residual pixels to obtain the amount of *person expected pixels* $P_x(S_m^n)$. Immediately after, we repeated the process of person segmentation and pixels count in depth image in order to obtain the *person acquired pixels* $P_a(S_m^n)$. Finally, we compared the amount of pixels counted in each image calculating the pixels rate P_a/P_x .

Similarly, we compare the amount of pixels obtained from the same person in different positions, but looking at the same direction.

4.3.2 Test Results

These results are the most important in terms of smart cameras integration. This experiment evaluates which cam-



(a) Different self occlusion cases (b) Front directions
Figure 4: (a) Rough partition of the FOV of a downward-looking 3D-camera in 3 regions R_n having a different behavior with respect of the self occlusion mechanism. Red dotted regions have a privileged exposition to the camera. (b) Four different front directions ϕ_m taken by the human target during the detection test.

Table 5: Human detection ratio between regions, relative to region 2

	Region 1	Region 3
Etron	1.02	1.18
Parrot	1.34	1.92
Intel	1.39	1.33
Asus	1.14	1.23

era fits best our needs to detect a person in a downward position. From these results, we inferred that the best camera in order to replace the Xtion Pro is the Etron with the better rate of human detection reliability. Even when the amount of pixels is lower than the Xtion, the 15% is enough to determine and to segment the head and the shoulders. We successfully integrated this camera in a counting and tracking smart camera system.

The Figure 5 presents the evaluation of the position and direction influence using the Xtion camera. This figure shows the segmented region size of a human target (in pixels), grouped by positions and direction. Systematically, we observed the self occlusion in the central position, for any direction. Regarding the camera networks, we obtained more pixels in the external positions of the image, where the overlapping regions between cameras are placed. In consequence,

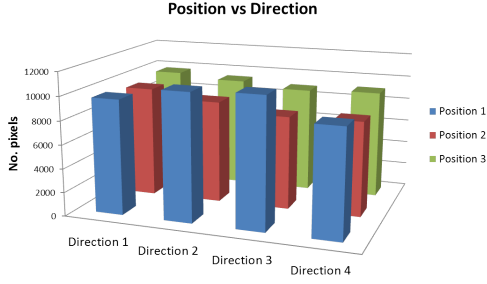


Figure 5: Acquired pixels per direction and position the target handover (the step where one camera pass the responsibility to another for tracking the person in the system) between cameras will be done with more pertinent data.

The Figure 6 presents depth images of the same person in the same position and front direction. We noticed that the Intel R200 had problems to detect the top of heads and the Parrot has a low resolution depth map (due to 10x10 pixels tiles) and has some difficulties to detect people.

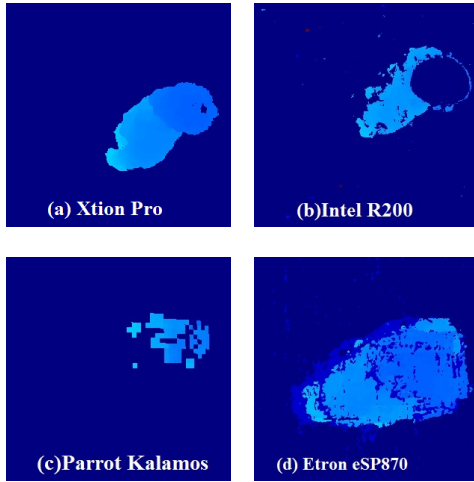


Figure 6: Different 300x300 pixels depth images of the same person. Invalid (no-data) pixels have dark blue color.

4.4 Implementation

In order to run these experiments, we created an I/O brick camera interface and a *fast camera characterization toolkit* (FCCT). All software is developed in C++. Our generic design uses an abstract model of camera and I/O interface functionality. This allowed the creation of a single test implementation, regardless of camera model, and despite different formats and image resolution. Equally importantly, we developed the user graphic interface to control the cameras and to modify the test parameters. Figure 7 presents the FCCT, I/O interface and GUI object-oriented architectures used to evaluate the 3D-cameras.

5. CONCLUSION

The interest of the stakeholders related to smart environments has been rapidly increasing in the last years. A large amount of 3D sensing research work has been produced by the academic community. At the same time, the global and local manufacturers, led by the large number of new products and solutions developed by the industry, were pushed to

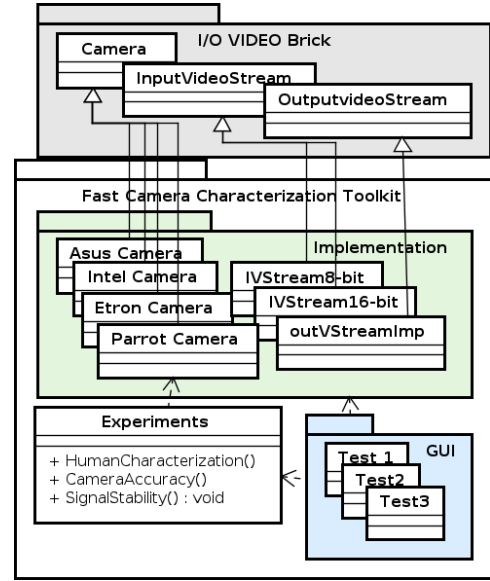


Figure 7: Class diagram of the FCCT architecture. develop their own 3D-sensing solutions. As a result, several new devices are coming to the market.

In this paper, we presented a set of properties characterizing RGB-D cameras. Each property was evaluated with a set of procedures that can be repeated and executed with any new depth sensor. We compared and evaluated a set of recently released cameras and presented the results of our analysis.

Additionally, we generated Table 6 assessing the pertinence of the use of each 3D-cameras in some main fields of applications, depending on each camera's properties.

We evaluated the usability of the generated depth maps in order to: detect patient body parts (*health*); detect gestures and then trigger actions (*gesture control*); robustly detect people in indoors (*room supervision*); provide pedestrian, cars and other obstacles detection (*driver assistance*); provide 3D point clouds for SLAM applications (*robotics*), record 3D multimedia content (*multimedia*), perform 3D image registration (*3D reconstruction*), support exterior applications influenced by sunlight (*Outdoors*), integrate a real-time and a limited computational resources solution (*embedded solutions*).

Our main contribution is the development of an evaluation framework platform for fast and early 3D-sensors characterization, focused on the integration into hybrid smart camera networks. This framework has a layered, object-oriented architecture intended for quantitative and qualitative analyses.

Based on [4], we expanded the hardware and software design and simplified the I/O interface architecture to accelerate the addition of new 3D-cameras to smart camera network systems. We improved camera integration by optimizing the software layer required to aggregate sensors, decreasing time-to-production and standardizing the communication between bricks. In table 6 we give our appreciation to each field of application according to the given results, especially to the smart cameras and integration in our system.

Finally we thank the Kalamos Parrot team for letting us test their camera.

Table 6: Cameras Applications Evaluation: ++ highly recommended , + recommended, 0 feasible, – not recommended, ––strongly not recommended

Applications	Cameras								
	Xtion	Kinect v2	SR300	UniBo	R200	eSP870	Parrot	PS4eye	ZED
Smart cameras	+	+	0	+	-	+	--	--	--
Health	+	+	+	+	+	+	0	--	+
Gesture Control	+	++	+	-	+	0	--	++	0
Room Supervision	+	+	0	0	0	+	--	--	+
Driver Assistance	-	0	0	+	0	0	++	-	++
Robotics	0	+	+	+	+	+	++	--	++
Multimedia	+	+	+	+	+	0	0	+	+
3D reconstruction	-	++	0	+	0	0	+	-	++
Outdoors	--	0	--	+	0	+	++	0	+
Embedded solutions	+	-	0	+	+	+	++	--	--

6. REFERENCES

- [1] Z. Aviv, D. Stanhill, D. Reif, R. Ziss, J. Danowitz, and E. Pertzov. Structured stereo. WO2015163995 A1, Oct. 2015.
- [2] S. Ben-Moshe, R. Kimmel, and M. Bronstein. Method and system for structured light 3D camera. US20120218464 A1, Aug. 2012.
- [3] K. Berger, S. Meister, R. Nair, and D. Kondermann. A state of the art report on kinect sensor setups in computer vision. In *Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications*, pages 257–272. Springer, 2013.
- [4] A. Burbano, S. Bouaziz, and M. Vasiliu. 3d-sensing Distributed Embedded System for People Tracking and Counting. In *Int. Conf on Computational Science and Comp. Intelligence*, pages 470–475, Dec. 2015.
- [5] H. Fu, H. Ma, and H. Xiao. Real-time accurate crowd counting based on RGB-D information. In *Int. Conf. on Image Processing*, pages 2685–2688, Sept. 2012.
- [6] D. F. Glas, D. Brscic, T. Miyashita, and N. Hagita. SNAPCAT-3d: Calibrating networks of 3d range sensors for pedestrian tracking. In *Int. Conf. on Robotics and Automation*, pages 712–719, May 2015.
- [7] M. Grzegorzek, C. Theobalt, R. Koch, and A. Kolb. *Time-of-Flight and Depth Imaging. Sensors, Algorithms and Applications*, volume 8200. Springer, 2013.
- [8] B. Khaleghi, S. Ahuja, and Q. J. Wu. An improved real-time miniaturized embedded stereo vision system (MESVS-II). In *Computer Vision and Pattern Recognition Workshops*, pages 1–8. IEEE, 2008.
- [9] D. Lefloch and et al. Technical foundation and calibration methods for time-of-flight cameras. In *Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications*, pages 3–24. Springer, 2013.
- [10] Q. Li, M. Biswas, M. R. Pickering, and M. R. Frater. Accurate depth estimation using structured light and passive stereo disparity estimation. In *Int. Conf. on Image Processing*, pages 969–972. IEEE, 2011.
- [11] J. Liu, G. Zhang, Y. Liu, L. Tian, and Y. Q. Chen. An ultra-fast human detection method for color-depth camera. *Journal of Visual Communication and Image Representation*, 31:177–185, 2015.
- [12] C.-C. Lu. Image process apparatus. US20150319425 A1, Nov. 2015.
- [13] S. Mattoccia. Stereo vision algorithms for fpgas. In *Proceedings of the Int. Conf. on Computer Vision and Pattern Recognition*, pages 636–641, 2013.
- [14] S. Mattoccia and P. Macri. A Real Time 3d Sensor for Smart Cameras. In *Int. Conf. on Distributed Smart Cameras*, page 29. ACM, 2014.
- [15] S. Mattoccia and M. Poggi. A passive RGBD sensor for accurate and real-time depth sensing self-contained into an FPGA. In *Int. Conf. on Distributed Smart Camera*, pages 146–151. ACM, 2015.
- [16] R. Nair, S. Meister, M. Lambers, M. Balda, H. Hofmann, A. Kolb, D. Kondermann, and B. Jahne. Ground truth for evaluating time of flight imaging. In *ToF and Depth Imaging. Sensors, Algorithms, and Applications*, pages 52–74. Springer, 2013.
- [17] D. Pagliari and L. Pinto. Calibration of Kinect for Xbox One and Comparison between the Two Generations of Microsoft Sensors. *Sensors*, 15(11):27569–27589, 2015.
- [18] M. Reisslein, B. Rinner, and A. Roy. Smart Camera Networks. *Computer*, 47(5):23–25, May 2014.
- [19] B. Rinner and T. Winkler. Privacy-protecting Smart Cameras. In *Int. Conf. on Distributed Smart Cameras*, page 40. ACM, 2014.
- [20] B. Rinner, T. Winkler, W. Schriebl, M. Quaritsch, and W. Wolf. The evolution from single to pervasive smart cameras. In *Int. Conf. on Distributed Smart Cameras*, pages 1–10. IEEE, 2008.
- [21] T. Teixeira, G. Dublon, and A. Savvides. A survey of human-sensing: Methods for detecting presence, count, location, track, and identity. *ACM Computing Surveys*, 5:427–450, 2010.
- [22] J. Thollot, X. Baele, and I. Ravysse. Stabilization of 3d Position Measurement. In *Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications*, pages 46–51. Springer, 2013.