This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2019.2915841, IEEE Transactions on Pattern Analysis and Machine Intelligence

Intel® RealSense™ SR300 Coded light depth Camera

Aviad Zabatani, Vitaly Surazhsky, Erez Sperling, Sagi Ben Moshe, Ohad Menashe, *Senior Member, IEEE,* David H. Silver, Zachi Karni, Alexander M. Bronstein, *Fellow, IEEE,* Michael M. Bronstein, *Fellow, IEEE,* Ron Kimmel, *Fellow, IEEE*

Abstract—Intel® RealSense™ SR300 is a depth camera capable of providing a VGA-size depth map at 60 fps and 0.125mm depth resolution. In addition, it outputs an infrared VGA-resolution image and a 1080p color texture image at 30 fps. SR300 form-factor enables it to be integrated into small consumer products and as a front facing camera in laptops and Ultrabooks™. The SR300 depth camera is based on a coded-light technology where triangulation between projected patterns and images captured by a dedicated sensor is used to produce the depth map. Each projected line is coded by a special temporal optical code, that enables a dense depth map reconstruction from its reflection. The solid mechanical assembly of the camera allows it to stay calibrated throughout temperature and pressure changes, drops, and hits. In addition, active dynamic control maintains a calibrated depth output. An extended API LibRS released with the camera allows developers to integrate the camera in various applications. Algorithms for 3D scanning, facial analysis, hand gesture recognition, and tracking are within reach for applications using the SR300. In this paper, we describe the underlying technology, hardware, and algorithms of the SR300, as well as its calibration procedure, and outline some use cases. We believe that this paper will provide a full case study of a mass-produced depth sensing product and technology.

Index Terms—Intel, RealSense, 3D Camera, SR300, Coded Light, Depth Reconstruction.

Fig. 1: Example of a 3D scan obtained with the SR300 camera.

1 Introduction

Just a few years ago, 3D depth sensing devices could be found only in a few academic and industrial labs due to their prohibitive cost. The manufacturers of such sensors catered mainly to high-precision applications such as metrology, industrial quality control, and digital heritage. High cost as well as large form factor precluded the mass use of depth sensors in commodity applications.

The past decade has witnessed the 'democratization' of 3D sensing technology and its transition to consumer applications. The first mass-produced low-cost depth sensor became accessible to consumers with the introduction of the Microsoft Kinect™, a depth sensor based on the structured light technology developed by Primesense®. Microsoft Kinect™ was the first commodity product 3D sensing device that enabled a new experience for gaming, where players could control and interact using body movement, without holding or wearing special devices.

The academic community has embraced the development of low-cost depth sensors, boosting research in the fields of computer vision and graphics. It was shown that complex problems of image understanding applications such as gesture and action recognition [1], [2], background segmentation, facial expressions analysis [3], marker-less motion capture [4], and 3D enhanced SLAM [5]–[7] are

All the authors were affiliated with Intel RealSense Group, Israel when this work was done. A. M. Bronstein and R. Kimmel are also with the Computer Science Department, Technion, Israel. M. M. Bronstein is with the Department of Computing, Imperial College London, United Kingdom and Institute of Computational Science, USI Lugano, Switzerland (e-mail: michael.bronstein@imperial.ac.uk).
Manuscript received September 20, 2018, revised December 18, 2018.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2019.2915841, IEEE Transactions on Pattern Analysis and Machine Intelligence

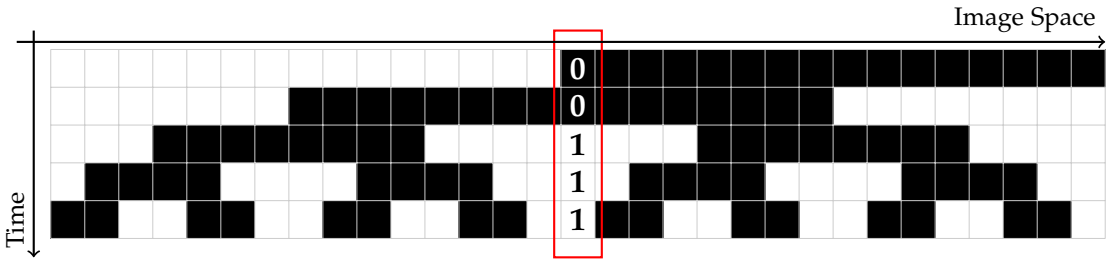


Fig. 2: 5 bit Grey code in time and image space.

greatly simplified by providing additional depth information. Even-though modern deep learning techniques allow nowadays purely 2D image-based solutions to some of the aforementioned problems [8], such methods require massive training datasets and are computationally expensive, which limits their applicability in real-time and power-efficient platforms such as mobile phones, and drones. Finally, the recent trend of 'geometric deep learning' [9], [10] tries to combine the best of the two worlds, developing deep neural network architectures capable of dealing with geometric data.

Popular optical depth sensing technologies includes passive methods [11] such as Shape-from-Shading and Shape-from-Motion and active methods such as active stereo, structured light [12], and Time-of-Flight (ToF) [13]. Intel® RealSense™ low cost coded light depth camera was motivated by academic prototypes originally developed in [14]–[16]. In 2014, Intel introduced the world smallest consumer depth camera under the retail name *F200*. Its successor SR300 was released the following year as an upgraded camera with better depth quality and lower cost. Coded light technology was chosen for its high image quality and resolution, ability to handle texture-less objects, low cost, small form factor, and mass production capability.

In this paper, we overview the technological foundations of the SR300 sensor. The rest of the paper is organized as follows: Section 2 presents the coded light technology used in SR300. Section 3 provides an overview of the SR300 hardware. Section 4 gives detailed description of the depth reconstruction pipeline. Section 5 presents the role of the firmware and some of the applications implemented in it. Section 6 reviews the optical calibration procedure of the SR300. Section 7 presents some use case examples and applications of the SR300, and Section 8 discusses its validation procedure and shows experimental reports.

2 Coded Light Technology

Triangulation-based *passive stereo* systems use two or more views (captured by respective cameras) of the same scene to determine the 3D position of the scene's objects. Pixel descriptors from these cameras are matched in order to calculate the depth of the pixels, given a known a priori calibration of the setup. Usually, the epipolar constraint is used to narrow the search space, but finding pixel correspondence is not an easy task especially when the object has little or no texture.

Structured light refers to a variety of *active stereo* techniques exploiting an additional controlled light pattern (or a series of patterns) projected onto the scanned scene, by replacing one of the sensors with a projector. Such patterns greatly simplify the correspondence problem by uniquely encoding each scene point thus allowing to identify corresponding points across views. *Coded light* refers to a subclass of structured light techniques that uses temporal or spatial codes to differentiate between projector points.

There are several types of patterns used in a coded light system. The difference is the system setup - monochromatic vs RGB sensors, 1D projector vs 2D, static vs moving scene, etc. The projector pattern determines the code each point will have.

Temporal Binary Gray Code

Temporal projected patterns were first introduced in 1982 [17]. Instead of projecting a single pattern, several patterns are projected sequentially and the "codeword" is constructed from this series of illuminations. The advantage of this kind of temporal patterns is that it can create dense and high resolution code. For example, in SR300 there are 245,280 pattern points, compared to single pattern structured light devices that typically project up to 30,000 points. In most cases, a 1D projector is used, since the depth can be calculated using the intersection between a ray and a plane. The main disadvantage of temporal coded light is its motion sensitivity, since the temporal coherency of the codes might break when objects in the scene move.

The temporal code can be RGB, gray-level, or binary. Every depth frame comprises a sequence of different projector patterns and image sensor "sub" frames. With temporal binary patterns, a point can either be illuminated or not in every sub-frame, thus creating a binary code. Binary codes typically require a longer pattern sequence to achieve spatial resolution compared to gray-level codes, but are simpler to compute and project.

Gray code [18] is a particular type of binary coding originally proposed in [19]. With this technique, the projected patterns are stripes images where each image has a different spatial frequency, so that eventually, each stripe in space is encoded using the Gray code created by all patterns in time, as depicted in Fig 2. With Grey code, there are no two transitions at the same location, which improves the actual spatial resolution for a given number of patterns.

3 Hardware Overview

The heart of the depth camera is the projector and the sensor employed for the triangulation process. The light wavelength band around 860nm was chosen to be in the near-infrared (NIR) range, a common choice in products that are interacting with humans and the light projection

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2019.2915841, IEEE Transactions on Pattern Analysis and Machine Intelligence

is intended to be invisible. There are two additional main components: the vision processor ASIC, on which the depth reconstruction pipeline is implemented, and the RGB sensor providing additional texture image, synced both in space and time to the depth image.

A major drawback of any triangulation-based system, including coded light, is its high sensitivity to mechanical and optical changes. In the SR300 design, special care was taken to ensure stability over time, temperature, external disturbances and lifetime of the product, while maintaining a small enough form factor so the camera could fit in small devices and Ultrabooks.

Mechanical assembly

The SR300 is packed in a small, $110 \times 12.5 \times 3.8\text{mm}$, $9.4\,\text{g}$ form factor. The main subassembly is a PCB, the motherboard of the SR300; all the components are connected through it. Beside the cameras, projector and the vision processor ASIC, the PCB also contains flash memory, color image signal processor (ISP), privacy led, and power delivery circuit. Additional subassembly components include leveling adhesive applied to the camera and projector subassemblies to create a level surface between both sub-assemblies, adhesive liner applied to the leveling adhesive and acting as the thermally and electrically conductive interface between the SR300 and the system assembly adhesive, and assembly frame, which adheres to the main subassembly and assists in rigidity, thermal dissipation, and EMI. The components arrangement is displayed in Fig 4.

IR Camera

The IR imaging sensor is a proprietary custom-designed sensor. It is a $1/6''$ 640×480 pixels global shutter CDS sensor, with frame rate of 600 fps (1200 fps in half resolution) and a 5T pixel size of $3.6\,\mu\text{m}$ optimized for high quantum efficiency at 860nm wavelength. A fixed focus lens with F-number of 1.9, field of view of 72 x 59 deg, and relatively high distortion (up to 15%) is used. The camera has an incorporated IR band filter.

Projector

The IR projector comprises a $860\,\text{nm}$ laser diode emitting a small spot laser beam, a lens expanding the beam into a vertical line, and a tiny micro electro-mechanical system (MEMS) mirror that moves horizontally to create the desired pattern. The MEMS rotates the mirror at high frequency

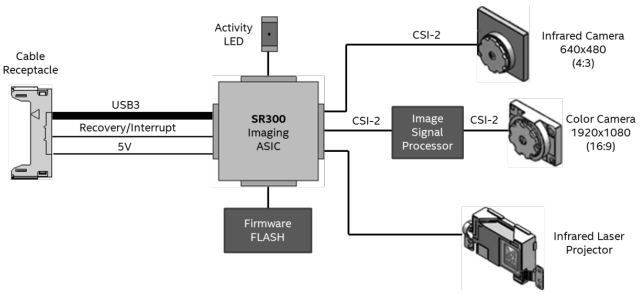


Fig. 3: SR300 3D imaging system.

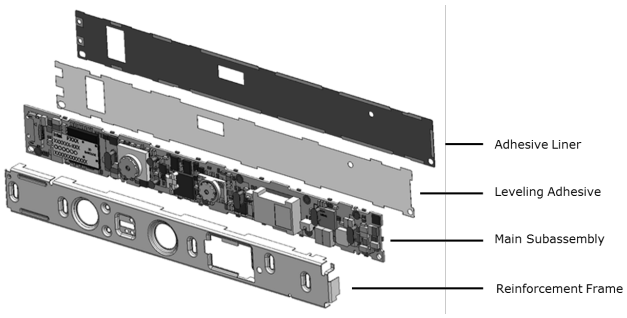


Fig. 4: SR300 assembly.

in harmonic motion to create an horizontal field of view of 72 degrees. It packs exceptional electro-static actuation, extremely low power consumption, and hardware safety mechanisms that provide protection from hacking and malfunctions, giving the SR300 Class 1 laser compliance. The shape of the projected pattern is controllable and is used to reduce artifacts [20]. The projected pattern is a vertical stripes image with the effect of the projector intrinsics: lens distortion, focal length, and optical axis.

RGB camera

The attached RGB camera is a full HD, 1920×1080 pixels rolling shutter sensor. The color camera lens has F-number of 2.4, field of view of 68 x 42 deg, and 1.5% distortion. The camera incorporates an IR cut filter so it is unaffected by the projected patterns. The RGB camera can work with HW level synchronization, giving the depth and the RGB image temporal synchronization well below $1\,\text{ms}$. The RGB camera is connected to a specialized ISP performing demosaicing, color correction, noise reduction, gamma correction, image sharpening, and format conversion to ensure a high quality color image.

3D Imaging ASIC

The vision processor ASIC implements the depth reconstruction pipeline described in Section 4. The custom design allows the pipeline to achieve a high frame rate while keeping the total system power consumption at around $1.9\,\text{W}$. The ASIC also incorporate a microcontroller for bootloading and device firmware. It supports USB-3 protocol to enable the high data rate of 60 fps VGA depth+IR image and 30 fps full HD color image simultaneously, MIPI protocol between the optical components, I2C, SPI UART, and JTAG for development.

4 Depth Reconstruction Pipeline

The Depth Reconstruction pipeline consists of two main modules. First, the Codeword Extraction module extracts the codeword for every pixel. It evaluates if every pixel is illuminated or not in every input frame. In addition, information about the sub-pixel location of the transitions in the pattern is calculated. Second comes the Depth Generation module, activated once all patterns are projected. It uses the codeword and the calibration data to reconstruct the depth at each pixel. In addition, it enhances the depth image. A flow chart of the depth reconstruction pipeline is depicted in Figure 5.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2019.2915841, IEEE Transactions on Pattern Analysis and Machine Intelligence

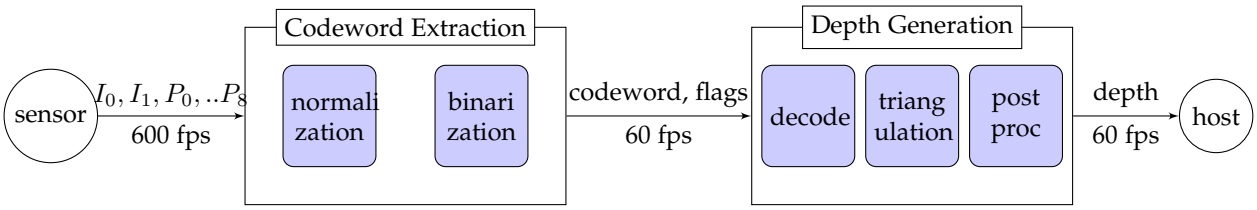


Fig. 5: Flow chart of the depth reconstruction pipeline.

CODEWORD EXTRACTION PHASE

The Codeword Extraction operates at the frame rate of the input images from the sensor (normally 600 fps) and its main goal is to evaluate the codeword per pixel. The Codeword Extraction pipeline consists of two main procedures: *Normalization* and *Binarization and sub-pixel estimation*, detailed below.

The input for this module are images of the Gray patterns. We used 9 gray code patterns denoted as $P_0, \ldots, P_8$ and two reference images: a fully illuminated image (denoted as I_1) and non-illuminated image (denoted as I_0). The projected sequence is $I_0, I_1, P_0, \ldots, P_8$. After substantial tuning and considerations of tradeoffs of the optical-algorithmical implementation, we found that 9 patterns offer the best performance. With 9 patterns, the scene is projected by a total of 511 transitions, which in full sensor resolution accounts for a transition every little more than 1 pixel, Which sets a limit for the depth spatial "lateral" resolution at approximately 1.25 pixels.

When a reference pattern I_0/I_1 is projected, it is stored in memory to be used later for normalizing the P_0–P_8 projected patterns. The output of the Codeword Extraction phase is a frame with per pixel 9-bit code-word and flags that indicate the validity of the pixel and the sub-pixel location of a binary transition, if there is one.

Normalization

This is the first block of the reconstruction pipeline. Its main goal is to evaluate the amount of IR projector emitted photons reflected from the scene object in each pixel on the pattern image P_i. In order to do that, the reference images I_0 and I_1 are used. The non-illuminated image I_0 is used as estimation of the ambient light. The illuminated image is used as an albedo estimation. When a reference pattern arrives to the Normalization block, it is being stored in memory, after a $2D$ low pass filter in the form of 3×5 median filter, since the albedo and the ambient light are approximately piece-wise smooth. Both reference images and the patterns are converted from their native image digital level to photoelectrons using the sensors QE factor.

Binarization and sub-pixel estimation

Binarization is the process of converting each pixel to one-bit indication of whether the pixel was illuminated (1) or not (0) in the given pattern. The binarization block inputs are the normalized pattern images, the filtered reference images, and a 4-bit state map. The state map is being modified by the block during the process of the binarization, and holds information regarding the validity of the pixel, and the sub-pixel location of a transition if such exist. The binarization output is a 1 bit pixel data and the updated state map.

The Binarization process is done by using a maximum likelihood estimation [21]. We used the following model to describe the signal,

$$y = \sigma^2 + \beta + \alpha x + \alpha^2(x^2\rho^2 + \eta^2) \tag{1}$$

where y is the signal sampled by the sensor, x is the signal reflected from the emitter projection, σ is the readout noise, ρ is the speckle noise coefficient, η is uncertainty term, α is the albedo, and β is the ambient light.

We characterize the sensor and projector to evaluate the noises (σ, ρ and η). η factor is not standard, and is used to compensate over other phenomena not being considered. The albedo and the ambient light, α and β, respectively, are evaluated using the reference images; $\alpha = I_1 - I_0$, $\beta = I_0$.

The binarization using (log) maximum likelihood solves the following problem:

$$\begin{aligned} \min_{x} \quad & \tfrac{(y-v)^2}{v} + \log v \\ \text{s.t.} \quad & v = \sigma^2 + \beta + \alpha x + \alpha^2(x^2\rho^2 + \eta^2) \end{aligned} \tag{2}$$

Since the above minimization problem is complex and expensive in hardware resources, we implemented it using templates; We constructed several 2D templates per pattern with size of 3x9, each template is a possible x from 2, representing the signal horizontal cross-section along the pixel. Using the albedo estimation, the ambient light estimation, and the evaluated noise factors, we can calculate v. We then find the most likely template that describes the signal by evaluating the cost of each v and picking the template that produced the lowest cost. The chosen template tells us if the pixel should be marked as 1 or 0, which is stored in the proper position of the codeword, according to the pattern type. The chosen pixel template also indicate if and where a transition occurred with 0.1 pixels precision.

Special considerations are taken for the hardware implementation of the $\frac{1}{v}$ and $\log v$ functions. They are implemented using two lookup tables (LUTs) and linear interpolation, using a custom floating point representation. By the location of the most significant bit of v the scale is determined, the 4 bits after the most significant are used as the entry to the lookup tables and the residual bits are used to increase the accuracy by multiplying the linear interpolation LUT result by the residual with the residual size as the scale for the interpolation. The accuracy of the function is up to a millionth of the result of the function. For

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2019.2915841, IEEE Transactions on Pattern Analysis and Machine Intelligence

example, dividing $v = 1950$, we have,

notation	description	Decimal	Binary
v	to be divided	1950	10111010101110
s	scale	13	**1**0111010101110
l	LUT entry	7	1**0111**010101110
r	Residual	174	10111**010101110**
den	Residual scale	9	10111**010101110**

$$\frac{1}{v} \cdot 2^{s+16} \approx LUT_{\text{div-op}}[l] + \left(r \cdot LUT_{\text{div-interp}}[l]\right) \gg den \quad .$$

The Binarization block also determines the confidence of the pixel by estimating the SNR. This is done by using the model in Equation 1 with the albedo and ambient information only. We compute the deviation of the signal when the emitter illuminates the pixel from $\sqrt{y_{x=1}}$ and deviation of the signal when the emitter does not illuminates the pixel from $\sqrt{y_{x=0}}$. Combining the two factors gives an uncertainty estimation. The confidence is the ratio between the uncertainty and the albedo,

$$\begin{aligned} v_p &= \sigma^2 + \beta + \alpha + \alpha^2(\rho^2 + \eta^2) \\ v_n &= \sigma^2 + \beta + \alpha^2\eta^2 \\ \sigma &= \sqrt{v_p} + \sqrt{v_n} \\ \text{confidence} &= \frac{\alpha}{\sigma}. \end{aligned} \tag{3}$$

We then apply a threshold to invalidate low confidence pixels. Later in the pipeline, there is an effort to fix invalid pixels if there is enough information in the pixels neighborhood.

Depth Generation Phase

The Depth Generation phase works at the frame rate of the output (normally 60 fps) and its main goal is to evaluate the codeword per pixel. The pipeline consists of three main modules: *Codeword decoding and correction*, *Triangulation*, and *Post-processing*.

Codeword Decoding and Correction

As stated before, Gray code is used to encode the projector points, so once the code word is extracted, it needs to be decoded into a binary code for ease of use. The conversion is done in the standard fashion as in the original Gray code patent [18].

The accuracy of the code has a dramatic effect on the quality of the depth image, since triangulation amplifies errors. Thus, it is crucial to purify the codeword. To do that, we perform complex filtering on the binary code. First, in the binary codeword representation, the image points are a horizontal monotonic ascending function. Another key observation is that in a coded light system, the information resides in vertical lines and hence horizontal transitions in the projected patterns. Simple filters that are usually applied to images, such as Gaussian or bilateral filter, do not perform well on the code images and usually cause artifacts, due to the special structure of the code. Several special code filters [22] are applied to the binary coded image, using a linear code assumption derived from the depth piecewise smoothness model. The code correction is done using three types of filters described below.

4.0.1 Vertical correction

The Vertical Correction is a non linear 2D filter that analyzes transitions misalignment along a vertical path. It is used to straighten vertical code transitions by refining the sub-pixel transition locations from corresponding transitions in vertical neighbors. Ideally, transitions in three subsequent sub-pixel points should lie on a line. The principle of operation is shown in Figure 6.

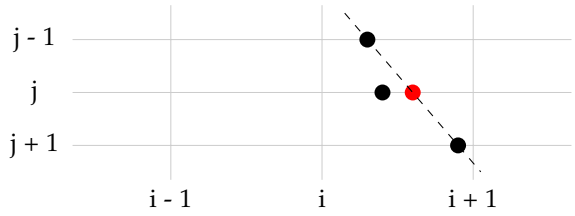


Fig. 6: Illustration of the operation of the vertical code filter. The original location of the detected transition in pixel (i, j) (in black) is corrected (red) to align with the transition in the rows above and below.

4.0.2 Horizontal correction

The Horizontal Correction is a non linear 2D filter that analyzes transitions misalignment along a horizontal path. The filter detects code transitions to the left and right of the current code transition in one line of code, and adjusts the sub-pixel position of the current transition such that they are as equally spaced as possible. The principle of operation is shown in Figure 7.

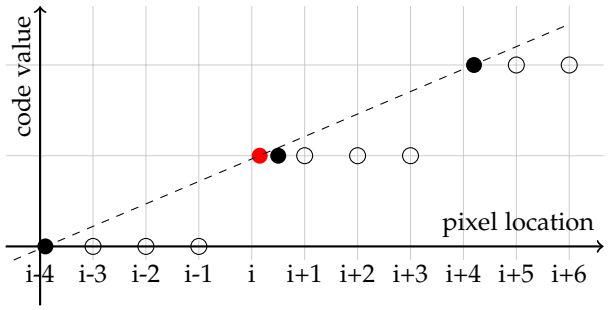


Fig. 7: Illustration of the operation of the horizontal code filter. Detected transitions are marked by black circles. The original location of the detected transition in pixel (i, j) is corrected (red) to align with the next and previous transitions.

4.0.3 Consistency filter

Consistency Filter is a one dimensional horizontal filter targeted to smooth the quantization of the code and to correct misinterpretation of bits in the code words by identifying valid small code segments, interpolate them to a higher resolution linear code, and extrapolate them where invalid or inconsistent code exist. First, segments of code are identified from one detected transition to the next. Second, the beginning and the end of the segment is interpolated through the segment to form a high-resolution code by re-sampling each of the pixels in higher precision than the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2019.2915841, IEEE Transactions on Pattern Analysis and Machine Intelligence

quantization of the code. In cases when only one of the boundaries is identified, the consistency filter extrapolates the code from the valid end. If both ends are invalid, the consistency filter invalidates the pixels in between, as is most likely, some artifact from motion or inaccurate code detection. The operation of the the filter is shown in Figure 8.

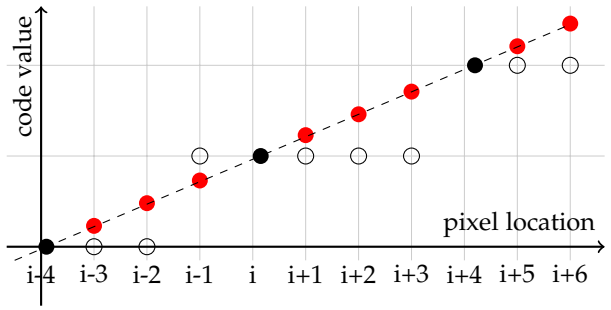


Fig. 8: Illustration of the operation of the consistency filter code filter. Detected transitions are marked by black circles. The code values between transitions are interpolated and mis-detected codes are fixed.

The aforementioned filters are applied in order to fix inaccuracies in the detection of the transitions by alternating horizontal and vertical filter several times (typically 2-3). Such a sequence of filters has the effect of using filters with larger support. Then, the consistency filter is applied to improve the accuracy of the non-transition pixels by relying on the corrected transitions. The result is a smooth and more accurate binary code map with distinguishable depth discontinuities.

Triangulation

Producing depth from the codewords and the calibration is a mathematical computation called *triangulation*, where the intersection between the rays from the sensor and the planes projected by the projector is calculated. The inputs are the pixel location in the sensor array, the codeword and the intrinsic and extrinsic parameters describing the system.

Camera model

We used the standard pin-hole camera model. Given a point X_w in homogeneous world coordinates, it is projected onto the camera image plane to homogeneous coordinates X_c by the following model

$$X_c = K_c \begin{bmatrix} I & 0 \end{bmatrix} X_w, \tag{4}$$

where K_c is the intrinsic camera matrix,

$$K_c = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}.$$

The camera system of coordinates is assumed to coincide with the world coordinate system, therefore trivial rotation and translation (extrinsic) matrices are imposed.

To account for the lens distortion, a parametric distortion model is applied. The corrected camera coordinates vector X_c is related to its uncorrected counterpart X'_c through the following **inverse** model,

$$X'_c = K_c D_\alpha \left(K_c^{-1} X_c \right), \tag{5}$$

where D_α is a plane to plane map given by

$$\begin{aligned} x' &= \left(1 + \alpha_1\rho^2 + \alpha_2\rho^4 + \alpha_5\rho^6\right) x + 2\alpha_3 xy + \alpha_4 \left(\rho^2 + 2x^2\right) \\ y' &= \left(1 + \alpha_1\rho^2 + \alpha_2\rho^4 + \alpha_5\rho^6\right) x + 2\alpha_4 xy + \alpha_3 \left(\rho^2 + 2x^2\right). \end{aligned} \tag{6}$$

with $\rho^2 = x^2 + y^2$. The parameters $\alpha_1, \alpha_2, \alpha_5$ determine the radial distortion component, while α_3 and α_4 account for the tangential one. We refer to the vector $\alpha = (\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5)^\top$, as the *camera inverse distortion* parameters. We emphasize the difference between this inverse model versus the standard forward model used in other systems. The benefits of this approach are simplification of the depth reconstruction arithmetics, better accuracy and spatial resolution as a result of avoiding interpolation required to re-align the pixel grid.

Projector model

The projector is modeled as a one-dimensional pin-hole system,

$$X_p = K_p \begin{bmatrix} R_p & t_p \end{bmatrix} X_w = P_p X_w, \tag{7}$$

where X_p is the 2×1 vector of homogeneous coordinate on the projector line, K_p is the 2×3 intrinsic matrix, and R_p, t_p are the extrinsic rotation and translation transformation parameters relating between the camera and the projector coordinate systems. We combine the intrinsic and the extrinsic projector parameters into a single 2×4 projection matrix P_p.

Projector distortion exists due to MEMS mirror electromagnetic motion and lens optics. Accounting for projector distortion is more complex than in the case of the image sensor lens due to the ambiguity of the second coordinate, which is unobserved. To cope with it, a proprietary mechanism is used [23]. The corrected projector coordinates vector X_p is related to its uncorrected counterpart X'_p through the following **forward** model

$$X_p = D^p_\alpha X'_p = D^p_\alpha \left(P_p X_w \right), \tag{8}$$

where D^p_α is a plane to plane map derived from the following 2D distortion function,

$$\begin{aligned} x' &= x \\ y' &= y + \alpha_1 + \alpha_2 x + \alpha_3 x^2 + \alpha_4 x^4 + \alpha_5 xy. \end{aligned} \tag{9}$$

Depth is reconstructed by observing x_p, the first coordinate of the two-dimensional projector location $X_p = (x_p, y_p)^\top$, at a camera pixel location X_c. Note that both X_c and X_p are after distortion, and y_p is unobserved. Reconstruction is performed by the inverse projection operator R,

$$X_w = R\left(X'_c, x'_p\right) \tag{10}$$

where $X'c = D_c^{-1}\left(X_c\right)$ and $x'_p = D_p^{-1}\left(x_p\right)$ are the undistorted camera and projector coordinates, respectively. Since the location of each pixel X_c is fixed, X'_c is precomputed. On the other hand, the projector coordinate cannot be undistorted as the inverse distortion D_p^{-1} requires the knowledge of the unobserved second coordinate y_p. Assuming x_p is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2019.2915841, IEEE Transactions on Pattern Analysis and Machine Intelligence

known, for a given X'_c, the reconstructed world coordinate is given by $X_w = R\left(X'_c, x'_p\right)$. Re-projecting the result on the projector plane yields $X_p = D_p(P_p X_w)$. In other words,

$$X_p = D_p\left(P_p R\left(X'_c, D_p^1\left(X_p\right)\right)\right) \tag{11}$$

is an identity on $\mathbb{R}^2$. Fixing the observed x_p, the identity restricted to $\mathbb{R}$ is

$$y_p = D_p\left(P_p R\left(X'_c, D_p^{-1}\left(x_p, y_p\right)\right)\right) = S\left(y_p\right) \tag{12}$$

The operator S is continuous and has a fixed point, to which the result of application of S multiple times to some initial y_p converge. In the absence of any better initialization, y_p is initialized to zero, resulting in a sequence of fixed point iterations $y_p = S \circ \cdots \circ S\left(0\right)$. In many practical situations when the distortion is mild, a single iteration suffices, giving $y_p \approx (0)$.

Performing the reconstruction in hardware accurately is done in a floating point arithmetic unit. Coefficients are precomputed using the calibration parameters described above to reduce the complexity of the computation. The floating point results are then clipped and converted to a 16-bit integer representing distances up to $8\,\mathrm{m}$, which yields the theoretical depth resolution of $0.125\,\mathrm{mm}$.

Depth post processing pipeline

At depth discontinuities (edges), the code estimation might be inaccurate since it is difficult to distinguish between projected pattern transition and a scene (albedo or geometry) transition. The outcome is typically noisy edges, especially in the presence of motion. Since edge sharpness is a key characteristic of depth image quality, post-processing filtering is typically applied to enhance the edges. Other key characteristics are the smoothness of objects in the scene and the minimal detectable object size. Typically, edge-preserving filters are used. We use a configurable series of post processing filters aimed to improve the mentioned characteristics [24].

The input to the pipeline is a 16-bit depth map and 4-bit flags that holds the state of the pixel (invalid, high confidence, etc.). The pipeline is depicted in Figure 9.

Gradient Filter

Gradient Filter improves the depth image by removing strong gradients, which mostly come from noise and motion artifacts. It is a 3×3 filter that calculate the gradients in all directions and if the magnitude of the gradient is higher than some depth dependent threshold, it invalidates the pixel. As depicted in Figure 9, there are actually two gradient filters, one at the beginning, and one at the end of the depth post processing. The second gradient filter is usually configured with more conservative thresholds, and mostly remove only a small amount of noise, that might have been added during the pipeline.

Morphological Filter

Morphological Filter is a 3×3 sliding window that works on the flags. It creates a $0/1$ map from the flags according to the pixels validity, and concatenates it to a binary number, which is then used to access a pre-configured LUT containing a set of 3×3 patterns indicating the validity of the central pixel. Examples of how it operates is shown in Figure 10.

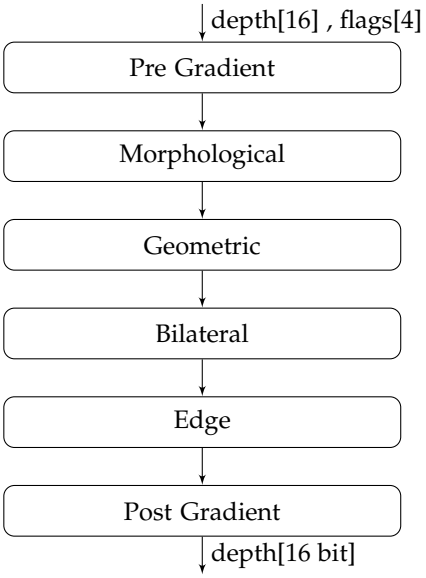


Fig. 9: Flow chart of the depth post-processing pipeline

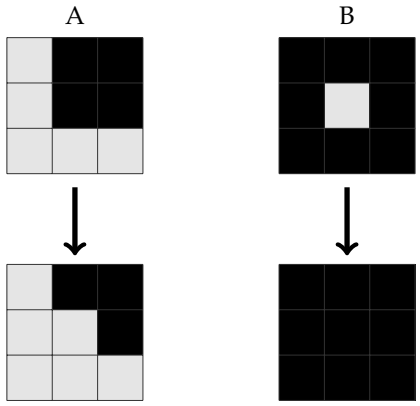


Fig. 10: Examples of the Morphological filter operation. (A) validating an invalid center pixel since it is part of well determined slope. (B) invalidating a spurious central pixel. (Light color denotes valid pixels).

Geometric Filter

A 5×5 filter that works on the flags and the depth image. It creates a $0/1$ map from the flags, and compare the 2D patch to a set of templates shown in Figure 11, and determine the pixel validity accordingly. For pixels whose binary neighborhood matching a template (with some high confidence), we mark the central pixel as valid/invalid based on the value of the central pixel in the template. In a case where a pixel is marked to be valid, we assign it the median depth value of his neighborhood masked by the matching template. This filter not only preserve edges, but takes into account the validity and the spatial location of the neighbors pixels.

Bilateral filter

For post-processing the depth image, we use a fixed-point implementation of the Bilateral filter [25]–[29] with two major modifications [30]. First, the spatial and radial smoothing

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2019.2915841, IEEE Transactions on Pattern Analysis and Machine Intelligence

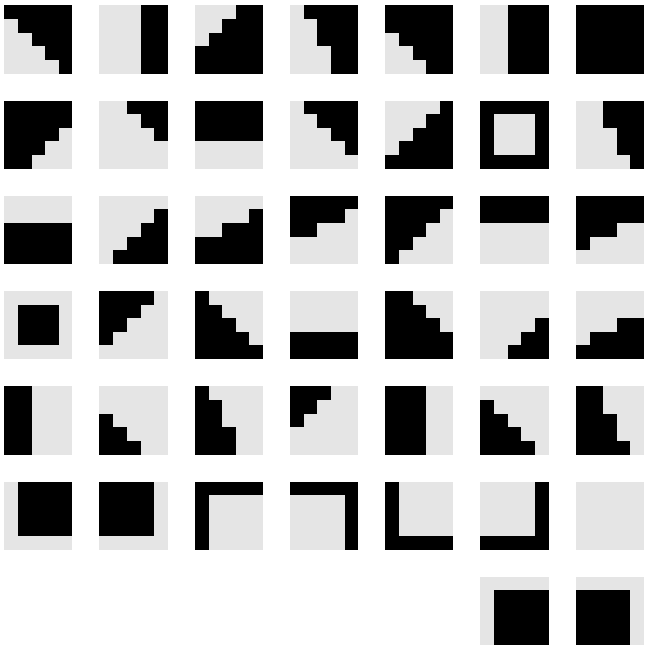


Fig. 11: The templates used by the Geometric filter (light color denotes valid pixels).

parameters, σ_s and σ_r respectively, are depth dependent and not constant throughout the image. This is due to the nature of the errors in triangulation based depth camera, where the error increases in proportion to the distance. Without depth adaptive approach, the bilateral filter configuration will result in oversmoothing of close objects and undersmoothing of distant ones [30]. Another important refinement is the pre-smoothing of of the depth central pixel, by a 3×3 box filter that averages the valid neighboring pixels with weighted confidence. The filter uses a 15×15 window.

Edge/Upscale filter

Edge filter is a 3×3 sliding window that works on the flags and the depth image. For each pixel it estimates the safest one dimensional interpolation by analyzing all the options of an edge going through the central pixel. Let x be the center pixel and $a_1, \ldots, a_4, b_1, \ldots, b_4$ its neighbors,

$$\begin{bmatrix} a_1 & a_2 & a_3 \\ b_4 & x & a_4 \\ b_3 & b_2 & b_1. \end{bmatrix}$$

The filter looks for the minimal delta pair (a_i, b_i) such that the values a_i, b_i are below some threshold. If this is the case, even when the pixel x is on an edge, this direction is safe to interpolate from the respective a_i, b_i. This way, we avoid averaging a pixel with values from different sides of an edge. The Edge filter can also be used for edge-preserving line duplication (vertical upscaling). This is useful for the 1200FPS half resolution image which is first upscaled with alternating invalid pixel rows, and then the edge filter is used to fill in the central pixel from directions a_1b_1, a_2b_2, a_3b_3 or a_4b_4 as in the normal case.

5 Firmware applications

As mentioned in Section 3, the SR300 vision processor ASIC is a system on chip with a CPU that runs the device firmware (FW). The FW is combined with an advance statistics gathering mechanism, capable of sampling the signals inside the depth reconstruction pipeline, provides the ability to create real-time autonomous applications that improves the usability and quality of the SR300 like auto exposure, laser intensity control, and MEMS control loop responsible for stabilizing the effective field of view . The FW also exposes depth controls enabling the user to change the characteristics of the depth map according to the specific need.

Depth controls and presets

The SR300 exposes the following main depth controls:

Filters option controls the amount of filtering and post processing done on the depth image. Low value is less processing. Value, 0, yields a *Skeleton mode*, where only high confidence code transition pixels are preserved.Value, 7, produces *blob mode*, where all available depth is shown and smooths out into blobs. Different filter options can be used per application, for example, high accuracy 3D scans may want to have only high confidence data, and long range application could wish to see as much as possible of the global structure without the fine details.

MvR (Motion vs Range) The longer the integration time of the sensor, the farther the camera can see, but more motion artifacts occur. This control enable the user to control this trade-off, according to the dynamics of his usage. The control simplifies the interaction with sensor exposure, FPS and switching between half-resolution upscaled image to full resolution mode. Small value means high FPS up to 120 and short integration time on half-resolution upscaled image while high numbers go to full resolution lower FPS.

Confidence. The validity of the pixel can be controlled using this property. There are several validity criteria, as mentioned in section 4, like SNR based criteria or pixels environment. Lower values will increase the amount of valid pixels in the image, enabling less *valid* pixels to go through.

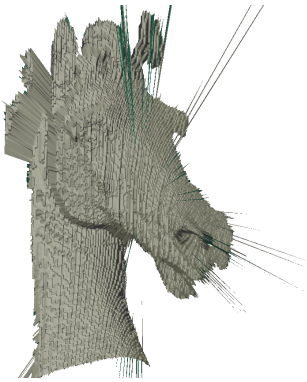
Accuracy controls the number of patters used in the generation of the code word. the more patterns are used, the better the spatial resolution is, but in long range the finer patterns may not be interpreted correctly.

RGB controls adjust the RGB camera image quality. These include sharpness, white balance, exposure, etc. by configuring the on chip ISP.

In addition to the above controls, the SR300 FW also expose a set of predefined configuration according to usage, which we call *Presets*. Instead of setting the controls individually, selecting a preset will set each control and additional FW based features like the *Auto range* automatically. For example, *Long Range* preset will set the confidence level to it's lowest, the accuracy to medium, MvR to 15 FPS and the filter to *blob mode*, which will enable the camera to work effectively up to 3 meters, and even see farther with more noise.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2019.2915841, IEEE Transactions on Pattern Analysis and Machine Intelligence

(a) Horse head model

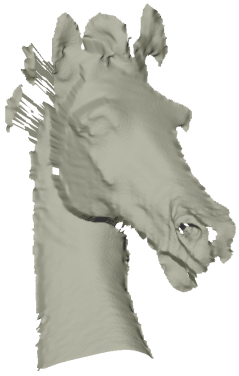


(b) No code correction, consistency filter and depth post processing

(c) With code correction, No consistency filter and depth post processing

(d) With code correction and consistency filter, No depth post processing

(e) With code correction and consistency filter, only gradient filter post processing



(f) With code correction and consistency filter and depth post processing

Fig. 12: Reconstructed depth of the horse model (12a) of size $20 \times 6 \times 15$ 1 cm in different stages of the pipeline

Auto-Range

The *Auto Range* is a real time FW process that automatically adjust the exposure and the laser power according to the scene [31]. As objects getting closer to the camera, or move faster, the exposure time of the sensor decrease, and in a certain point the laser power as well, to avoid saturation and to be able to perceive high velocity movements, and as objects are getting far from the camera, the laser power and exposure increases. This improves the quality of the depth image and enable usages like hand gesture recognition in large distances.

Power Gears

Power Gears are a real time FW process that given a set of rules, changes the operating mode of the camera and its power mode. A common used power gear is the *wake up* option, where the camera wakes ups and start streaming when a person is near the camera. This enables the SR300 to serve as a wake -up device for the operating system, like proximity sensor, but unlike proximity sensor, the set of rules can be more flexible. Additional power gears allow the camera to change FPS according to the dynamics of the scene, thus preserving power while still catching user gestures, etc.

MEMS FOV control

The MEMS FOV control loop is a real time FW process that makes small changes to the electric load of the mirror in order to make the opening angle of the mirror constant over time and temperature. It does it by sampling the very noisy location of the mirror in several points in its track, and detect slow or abrupt movement by applying advance control algorithms, thus keeping the projector FOV constant, which highly improves the image quality.

Thermal compensation control

The thermal compensation control is a real time FW process that makes small changes to the calibration parameters used in the triangulation process according to the temperature of the sensors [32]. Instead of using a power consuming, expensive hardware to keep the modules temperature, a parametric model is used to determine the correction to the initial parameters, in an open loop fashion, where there is no feedback from the camera. The correction itself is calibrated over a large batch of units, by measuring the behavior of the units in different temperatures, and its magnitude is dependent on the initial calibration of each unit. In average, the units depth accuracy is increased by up to a factor of two over the range of operating temperatures (10-70 °C).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2019.2915841, IEEE Transactions on Pattern Analysis and Machine Intelligence

6 CALIBRATION

The Calibration procedure is performed to calculate the *intrinsic parameters* (lens parameters of IR camera, IR projector, RGB camera) and the *extrinsic parameters* (Euclidean transformation of relative positioning and orientation relative to the IR camera). This evaluation procedure requires high accuracy and is performed per unit.

It is a common practice in most triangulation based depth camera to use a multi-view approach presented by Zhang [33] which includes presenting the camera with a planar target with detectable features, typically, a checker board, captured at a set of orientations. This approach has several major disadvantages for a high volume production line of coded light systems. First, the need to automatically capture several views of the target increases the acquisition time and requires complicated mechanics. Second, the feature detection quality may be greatly compromised by the low MTF of the IR sensor, affecting the quality of the entire process. Therefore, for the calibration of the SR300 a novel approach was developed. Single shot calibration using a target including a set of different plane geometries, without moving parts and not relying on feature detection [34]. This approach enables calibration with increased accuracy even for low quality sensors and optics, while using a single view calibration process. It enables to reduce calibration time and complexity in the production line to increase the volume and reduce the cost. To that end, a single-view feature-less depth and texture calibration using a target with known geometry of seven planes colored with high-contrast patterns was used. Instead of capturing multiple different views of a target of a single plane geometry, we use a target with seven different plane geometries. Each plane is textured with a checkerboard pattern, with a known and accurate geometry. The seven slanted planes include six different planar patches and a background plane. The target scene is shown in Figure 13.

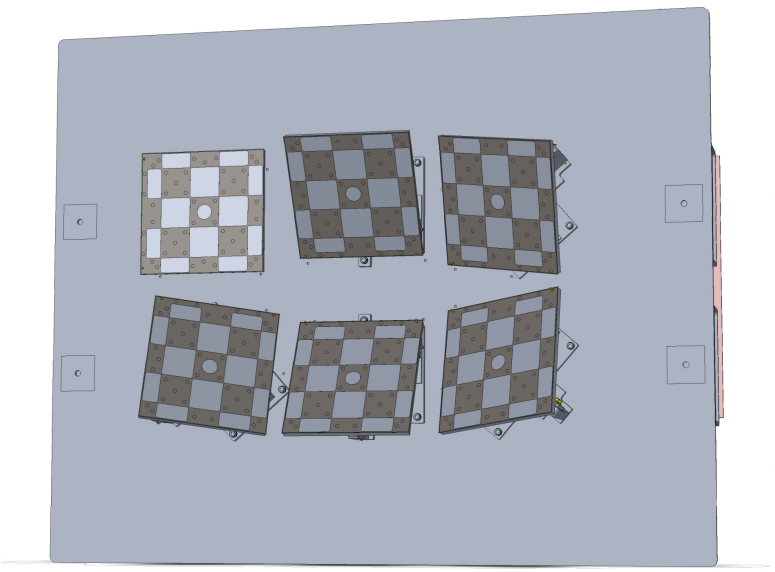


Fig. 13: Seven planar targets used for single-view featureless calibration of SR300

The calibration process is comprised of three stages.

Stage 1: Rough calibration process

An image of the target with uniform illumination is captured by the IR camera along with several images of a known projector patterns, which are projected on the target. Synthesized Images of the target and the same projected pattern as would have been seen by the system with current parameters are generated. Nominal production values are used as a rough initialization of the camera and projector intrinsics, the projector extrinsic, and the viewpoint (relative location and orientation) of the device with respect to the target, under assumption that the geometry of the target to be precisely known. The actual captured images will have some misalignment with the synthesized images. The calibration parameters can be determined to align the captured image to the corresponding synthesized image, by matching against their actually captured counterparts, and the system intrinsic and extrinsic parameters as well as the viewpoint are updated to produce the best match in the sense of un-normalized correlation error criterion. This first stage works entirely in the camera image plane using roughly all the pixels in the image to perform calibration, as opposed to the feature-based approaches that rely on a few tens of landmark points.

Stage 2: Depth refinement

An image of the projected code is further acquired by decoding a sequence of patterns projected by the projector. The depth image is reconstructed by triangulation using the current system parameters as evaluated in the first stage, and is matched against a calculated range image of the known target that would be seen by the camera with current intrinsics from the current viewpoint. The images are matched minimizing a geometric error criterion - squared Euclidean distance. The second stage works in the world coordinates, minimizing the actual geometric error of the reconstructed range image, and is also free of any use of landmark points.

Stage 3: RGB camera calibration

At the third stage, the intrinsics and extrinsic of the IR camera and projector, and the viewpoint of the entire system are fixed, and the intrinsic and extrinsic calibration of the RGB camera is performed. An image of the target with visible light illumination is acquired by the RGB camera, and is matched against a synthesized image of the known target that would have been seen by the RGB camera with current parameters. The two images are matched and the RGB camera parameters are updated to achieve the best match in the sense of radiometrically corrected correlation error criterion.

Calibration performance

We used this approach on millions of units. The calibration converged to produce valid results on more than 99.7% of the units, with average accuracy of 0.5% on 60 cm where it is being tested in the factory. The calibration process usually takes 30 s, and support two units in parallel, allowing production throughput of more than 100,000 units per tester per month.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2019.2915841, IEEE Transactions on Pattern Analysis and Machine Intelligence

OEM Calibration

This approach was also extended to an OEM Calibration procedure - where we recalibrate the unit after it was assembled in the OEM product, to compensate over changes to the module due to the assembly process. The OEM calibration uses just a single plane with checkerboard target, only to refine parameters that may have been changed in the assembly process, like the extrinsic between the projector and the IR camera, or the extrinsic between the RGB camera and the IR camera, while fixing all other parameters.This process achieves almost perfect yield, with accuracy of up to 3% in $2.5\,\mathrm{m}$.

7 Use Cases and Applications

SR300 was released with an extended SDK that enables developers to integrate advanced user interface into the applications, like hand tracking, gesture recognition and facial expressions. A number of games were also released as a proof of concept for the technology and its abilities. In addition, the SR300 was integrated in several products and was approved as a Windows hello^{TM} certified camera, which enables the user to login to his Windows based PC with ones face. Here we provide some examples of usages and applications.

7.1 Hand tracking and Gesture recognition

The camera tracks the hand and detects the full 3D skeleton of the hand, including all 22 joints, fingers information, gestures, and more. It can track one or two hands, providing precise joint-level locations and positions, identify gestures, which are certain significant hand postures or motions, for example a wave, tap or thumbs-up sign. Using this module, developers can enable their application to be controlled by hand motions, using visual cues alone (without a touch interface). For instance, interpreting a hand tap as selection.

7.2 Cursor Mode

The Cursor Mode provides real-time 3D motion tracking of the hand as a whole, making the hand act as the cursor. It can track one or two hands, provides fast and accurate tracking to follow the position of the hands in space, and converts the received data to smooth and responsive hand cursor data. This information is used to create a representation of the cursor on the monitor that can replace the mouse cursor.

7.3 Facial analysis

The face tracking and analysis module provides a suite of the following face algorithms.

Face detection locates a face (or multiple faces) from an image or a video sequence, and returns the face location in a rectangle. This feature can be used to count how many faces are in the picture and find their general locations.

Landmark detection further identifies the feature points (eyes, mouth, etc.) for a given face rectangle. The eye location is of particular interest for applications that change display perspectives based on the locations in the screen users are looking at. Other feature points can be useful to create a face avatar or find the orientation of the head.

Pose detection estimates the face orientation.

Expression detection calculates the scores for a few supported facial expressions such as eye-closed and eyebrow turning up.

Face recognition compares the current face with a set of reference pictures in the recognition database to determine the user's identification.

Pulse estimation tracks subtle change in face skin color over time and estimates the person's pulse rate.

Gaze tracking traces the user's eye movement and provides estimated eye gaze location on the display screen and angles from the origin.

7.4 Background Segmentation

The User Segmentation module generates a segmented image per frame which can be used to remove or replace portions of the image behind the user's head and shoulders (background). The module works on the synchronized color and depth images, and creates the segmented image in the resolution of color image that contains a copy of the input color data and a synthesized alpha channel (mask). Pixels which correspond to the background will contain an alpha channel of zero, and pixels which correspond to the user will have an alpha value greater than zero. Amazon Echo look^{TM} personal assistant device is using SR300 depth camera primarily for bokeh like effects using background blurring.

7.5 3D object Scanning

The 3D Scan module reconstructs the shape and appearance of stationary objects from a sequence of images taken from various viewing angles. Using segmentation and tracking techniques, the image sequence is converted into a 3D triangle mesh (e.g. PLY), for the purpose of simulation, editing/printing or analysis.

8 Experimental Results

The SR300 has being tested in a large variety of commercial applications (Section 7) as well as many academic works (see, e.g. [2]–[4]) and proved its capabilities in real time application. The mass production and availability of the SR300 has spawn several academic benchmark of the camera in different scenarios [35]–[37].

8.1 Validation Procedure

The production of the camera is being monitored by Intel quality assurance strict procedures: stress tests, power and heat cycles, drop test are part of the procedures the product is tested before granted a "ready for market" approval, to ensure the functionality of the camera over its entire lifespan. The camera is also tested in extreme condition such as temperatures, air pressure and moister, to verify its functionality and depth quality remains within the production specification for any approved condition.
In addition, each "bare bone" camera is tested individuality in the factory, ensuring it meets the depth, IR and RGB

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2019.2915841, IEEE Transactions on Pattern Analysis and Machine Intelligence

image quality standards. Among the tested criteria are: global accuracy, temporal and spatial noise, pixel validity percentage, planarity of planar objects, depth to RGB mapping, IR and RGB image quality metrics such as SNR and MTF. A camera is only shipped if all the metrics are positive, thus ensuring it meets the production specifications.

8.2 Characterization

The validation procedure is meant to ensure each unit is within spec. Every product release (Hardware or Firmware) is accompanied by a characterization procedure done on tens of units to measure the actual performance of an average camera. In this procedure, each depth metric is measured over all the effective depth range. Table 1 shows the results of the depth related metrics for the latest SR300 release. The target is a plain wall with 90% reflectivity and 80% of the image is taken into account. Global accuracy measures the median of the ground truth depth versus the calculated depth over all the pixels. For spatial noise plain fit RMS over 25×25 window is calculated, and the median of all the windows is taken. Temporal noise is the average STD of a pixel in 10 frames. Max Error is the 98th percentile of pixels absolute distance from the plane. Fill factor is the percentage of pixels that are valid and with error of less than 1% of the distance from the plane.

TABLE 1: Characterization report for latest SR300 package, shown at different distance from the camera.

Criterion/distance [*mm*]	500	700	900	1100	1300	1500
Global Accuracy [*mm*]	0.5	2.2	4.5	6.7	9.0	10.1
Spatial Noise [*mm*]	2.6	5.5	9.9	15.6	22.4	30.7
Temporal Noise [*mm*]	0.13	0.27	0.52	0.85	1.34	2.10
Max Error [*mm*]	3.9	9.2	16.8	26.2	37.3	50.2
Fill Factor [%]	98.2	96.4	90.9	85.3	78.8	69.7

Beside the simple depth metrics listed above, the units undergo various test to measure the depth sharpness, deformations, and distortion using complicated 3d models, and IR and RGB advance image quality analysis as part of the characterization procedure.

9 CONCLUSIONS

Depth sensing has undergone a revolution in the past decade, with new technologies such as MEMS pico projectors enabling the development of small form factor and low-cost depth cameras. Intel RealSense SR300 camera was the first mass-produced sensor that had the form factor, power, and price point allowing its integration into mobile devices. This, in turn, has enabled a new line of products, and improved existing products by supplying additional vision capabilities.

Nowadays, low-cost depth cameras are an active academic and industrial research field. Overall, we can observe a general trend of shifting towards time-of-flight (ToF) technologies. While ToF systems have multiple key advantages over triangulation based systems, current ToF solutions are still more expensive and have lower spatial resolution compared to the SR300. A new line of depth cameras from RealSense is based on stereo technology providing outdoor depth sensing capabilities at an even lower cost.

ACKNOWLEDGMENT

The authors thank the Intel RealSense group for all their efforts and creativity invested in the design, testing, and manufacturing of the SR300 camera.

REFERENCES

[1] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *Proc. CVPR*, 2011.

[2] A. Wetzler, R. Slossberg, and R. Kimmel, "Rule of thumb: Deep derotation for improved fingertip detection," in *Proc. BMVC*, 2015.

[3] H. Li, L. Trutoiu, K. Olszewski, L. Wei, T. Trutna, P.-L. Hsieh, A. Nicholls, and C. Ma, "Facial performance sensing head-mounted display," *TOG*, vol. 34, no. 4, p. 47, 2015.

[4] T. Weise, S. Bouaziz, H. Li, and M. Pauly, "Realtime performance-based facial animation," *TOG*, vol. 30, no. 4, p. 77, 2011.

[5] F. Steinbrücker, J. Sturm, and D. Cremers, "Real-time visual odometry from dense RGB-D images," in *Proc. ICCV Workshops*, 2011.

[6] C. Audras, A. Comport, M. Meilland, and P. Rives, "Real-time dense appearance-based SLAM for RGB-D sensors," in *Proc. Australasian Conf. Robotics and Automation*, 2011.

[7] C. Kerl, J. Sturm, and D. Cremers, "Dense visual SLAM for RGB-D cameras," in *Proc. IROS*, 2013.

[8] R. A. Güler, G. Trigeorgis, E. Antonakos, P. Snape, S. Zafeiriou, and I. Kokkinos, "Densereg: Fully convolutional dense shape regression in-the-wild." in *Proc. CVPR*, 2017.

[9] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: going beyond euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017.

[10] F. Monti, D. Boscaini, J. Masci, E. Rodolà, J. Svoboda, and M. M. Bronstein, "Geometric deep learning on graphs and manifolds using mixture model CNNs," in *Proc. CVPR*, 2017.

[11] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge University Press, 2003.

[12] D. Scharstein and R. Szeliski, "High-accuracy stereo depth maps using structured light," in *Proc. CVPR*, 2003.

[13] F. Remondino and D. Stoppa, *TOF range-imaging cameras*.

[14] A. M. Bronstein, M. M. Bronstein, E. Gordon, and R. Kimmel, "High-resolution structured light range scanner with automatic calibration," Tech. Rep. CIS-2003-06.

[15] O. Rubinstein, Y. Honen, A. Bronstein, M. Bronstein, and R. Kimmel, "3D-color video camera," in *Proc. ICCV Workshops*, 2009.

[16] R. Kimmel, "Three-dimensional video scanner," Patent US 7,756,323, 2010.

[17] J. L. Posdamer and M. D. Altschuler, "Surface measurement by space-encoded projected beam systems,"

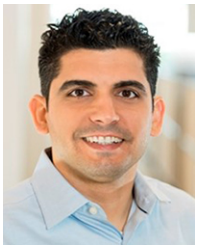
Computer Graphics and Image Processing, vol. 18, no. 1, pp. 1–17, 1982.
[18] G. Frank, "Pulse code communication," Patent US 2,632,058, 1953.
[19] K. Sato and S. Inokuchi, "Three-dimensional surface measurement by space encoding range imaging," *J. Robotic Systems*, vol. 2, pp. 27–39, 1985.
[20] V. Surazhsky, R. Kimmel, A. Bronstein, M. Bronstein, E. Sperling, and A. Zabatani, "Facilitating projection pre-shaping of digital images at computing devices," Patent US 9,792,673, 2017.
[21] A. Bronstein, A. Zabatani, R. Kimmel, M. Bronstein, E. Sperling, and V. Surazhsky, "Systems, methods, and apparatuses for implementing maximum likelihood image binarization in a coded light range camera," Patent US 9,952,036, 2018.
[22] V. Surazhsky, M. Bronstein, A. Bronstein, R. Kimmel, E. Sperling, A. Zabatani, O. Menashe, and D. H. Silver, "Code filters for coded light depth acquisition in depth images," Patent US 9,792,671, 2017.
[23] A. Bronstein, A. Zabatani, M. Bronstein, R. Kimmel, E. Sperling, and V. Surazhsky, "Projector distortion compensation in structured light depth reconstruction," Patent US 9,824,461, 2017.
[24] D. H. Silver, M. Bronstein, A. Bronstein, R. Kimmel, E. Sperling, V. Surazhsky, A. Zabatani, and O. Manashe, "Morphological and geometric edge filters for edge enhancement in depth images," Patent US 9,852,495, 2017.
[25] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. ICCV*, 1998.
[26] R. Kimmel, R. Malladi, and N. Sochen, "Images as embedding maps and minimal surfaces: movies, color, and volumetric medical images," in *Proc. CVPR*, 1997.
[27] N. Sochen, R. Kimmel, and A. M. Bruckstein, "Diffusions and confusions in signal and image processing," *JMIV*, vol. 14, no. 3, pp. 195–209, 2001.
[28] N. Sochen, R. Kimmel, and R. Malladi, "A general framework for low level vision," *IEEE Trans. Image Proc.*, vol. 7, no. 3, pp. 310–318, 1998.
[29] R. Kimmel, R. Malladi, and N. Sochen, "Images as embedded maps and minimal surfaces: movies, color, texture, and volumetric medical images," *IJCV*, vol. 39, no. 2, pp. 111–129, 2000.
[30] M. Bronstein, Z. Karni, A. Bronstein, R. Kimmel, E. Sperling, A. Zabatani, and V. Surazhsky, "Device and method for depth image dequantization," Patent US 9,940,701, 2018.
[31] A. Zabatani, E. Sperling, O. Mulla, R. Kimmel, A. Bronstein, M. Bronstein, D. H. Silver, O. Menashe, and V. Surazhsky, "Auto range control for active illumination depth camera," Patent US 9,800,795, 2017.
[32] A. Zabatani, S. Bareket, O. Menashe, E. Sperling, A. Bronstein, M. Bronstein, R. Kimmel, and V. Surazhsky, "Online compensation of thermal distortions in a stereo depth camera," Patent US 9,813,692, 2017.
[33] Z. Zhengyou, "Flexible camera calibration by viewing a plane from unknown orientations," in *Proc. ICCV*, 1999.
[34] A. Bronstein, A. Zabatani, R. Kimmel, M. Bronstein, E. Sperling, and V. Surazhsky, "Single view featureless depth and texture calibration," Patent US 9,794,545, 2017.
[35] M. Carfagni, R. Furferi, L. Governi, M. Servi, F. Uccheddu, and Y. Volpe, "On the performance of the intel sr300 depth camera: metrological and critical characterization," *IEEE Sensors Journal*, vol. 17, no. 14, pp. 4508–4519, 2017.
[36] R. House, A. Lasso, V. Harish, Z. Baum, and G. Fichtinger, "Evaluation of the Intel RealSense SR300 camera for image-guided interventions and application in vertebral level localization," in *Proc. SPIE Medical Imaging*, 2017.
[37] A. Burbano, M. Vasiliu, and S. Bouaziz, "3D cameras benchmark for human tracking in hybrid distributed smart camera networks," in *Proc. Distributed Smart Camera*, 2016.

Aviad Zabatani (B.Sc. in Electrical Engineering and Computer Science, Tel Aviv University) is a Senior Algorithms Engineer at Intel RealSense, Israel, where he has led the development of the F200 and SR300 Intel RealSense depth cameras. Among his research interests and expertise are 3D reconstruction, camera calibration, and real-time algorithms in image and signal processing. He authored several patents in the field of 3D imaging and calibration.

Vitaly Surazhsky is a Senior Algorithms Engineer at Intel RealSense, Israel. He received his PhD in Computer Science from the Technion in 2003. His main interests are image and geometry processing and computer graphics. He has published a dozen of papers and co-authored 10 granted patents.

Erez Sperling (B.Sc in Mathematics and Computer Science, Technion) has 32 years of engineering experience in the field of SW, VLSI, Computer vision, Cellular, CMOS imager, Image processing and 3D camera technology. Throughout his career, he has worked at Intel for over 15 years in total. He also held lead positions in the domain of Algorithm, ASIC, Architecture and VP R&D in Zapex, Mobilian, Emblase Semi, Advasense, Zoran, Inomize. Erez has authored over 20 patents in field related to 3D imaging.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2019.2915841, IEEE Transactions on Pattern Analysis and Machine Intelligence



Sagi Ben Moshe is Vice President, Emerging Growth Incubation (EGI) Group at Intel and Senior Vice president, Sensor Technologies at Mobileye. EGI is responsible for identifying, incubating and accelerating potential growth businesses for Intel. As part of EGI, Ben Moshe is also the General Manager of the RealSense Group, which develops next-generation hardware and software products that allow devices and machines the ability to perceive their surroundings in 3D. Ben Moshe oversees a multidisciplinary team including R&D, Marketing, Finance and Sales and is responsible for taking new technologies from conception to production, including product definition, budgeting and long-term planning. The introduction of the first Intel RealSense technology-enabled camera in 2015 earned him an Intel Achievement Award. Ben Moshe holds B.Sc. and M.Sc. degrees in Computer Science, both from the Technion. He has been granted multiple patents for his inventions.

Ohad Menashe received his M.Sc in electrical engineering From Tel-Aviv university. He has work experience in fields of 3d imaging hardware, signal processing, and calibration, and the autor/co-author of several dozen patents. Menashe is a Senior Member of IEEE, and is currently holds a position as a principal engineer at Autonomous Intelligent Driving. Previously, he was part of the Intel RealSense team and led the development of future Realsense product. His research interests includes optimization methods, signal processing, computer vision and Autonomous platforms.

David H. Silver studied Math & Biology at the Technion, Israel under the auspices of the Technion Excellence Program. During 2014-2016, he was a Microsoft Research PhD Scholars. He published several papers on genetics in Nature and PNAS and is a coauthor of several patents in the fields of signal processing and computer vision.

Zachi Karni received his Ph.D. degree in Computer Science from the Technion in 2004. He was a post-doc at the Max-Planck-Institut fr Informatik, Germany. He is an applied researcher with main interest in computer vision, graphics, geometry processing, robotics, and machine learning. He currently holds a research engineering position in Snapchat, and had served as a researcher at Intel RealSense. He also served as a senior researcher at HP Printing Automation Lab and MediGuide (a St. Jude Medical Division). Dr. Karni was an Adjunct Lecturer in Computer Science at the Technion. He is a member of the ACM SIGGRAPH community and has chaired the SIGGRAPH Israel Local Chapter.

Alexander M. Bronstein is a faculty in the Department of Computer Science at the Technion Israel Institute of Technology and a Principal Engineer at Intel. His research interests include numerical geometry, computer vision, and machine learning. Prof. Bronstein has authored over 100 publications in leading journals and conferences, over 30 patents and patent applications, the research monograph "Numerical geometry of non-rigid shapes", and edited several books. Highlights of his research were featured in CNN, SIAM News, Wired. Prof. Bronstein is a Fellow of the IEEE for his contribution to 3D imaging and geometry processing. In addition to his academic activity, he co-founded and served as Vice President of technology in the Silicon Valley start-up company Novafora (2005-2009), and was a co-founder and one of the main inventors and developers of the 3D sensing technology in the Israeli startup Invision, subsequently acquired by Intel in 2012. Prof. Bronstein's technology is now the core of the Intel RealSense 3D camera integrated into a variety of consumer electronic products. He is also a co-founder of the Israeli video search startup Videocites where he serves as Chief Scientist.

Michael M. Bronstein (PhD 2007, Technion, Israel) is a professor at Imperial College London, UK. and USI Lugano, Switzerland and Principal Engineer at Intel. During 2017-2018 he was a Radcliffe Fellow at the Institute for Advanced Study at Harvard University. He has held visiting appointments at Stanford, MIT, Harvard, TUM, and Tel Aviv University. Prof. Bronstein's main research interest is in theoretical and computational methods for geometric data analysis. He authored over 150 papers, the book Numerical geometry of non-rigid shapes (Springer 2008), and holds over 30 granted patents. He was awarded four ERC grants, two Google Faculty Research awards, Amazon AWS Machine Learning award, Facebook Computational Social Science award, Rudolf Diesel fellowship, Royal Society Wolfson Merit award, and Dalle Molle Prize. Prof. Bronstein is a Fellow of IEEE and IAPR, alumnus of the Technion Excellence Program and the Academy of Achievement, ACM Distinguished Speaker, member of the Young Academy of Europe, and World Economic Forum Young Scientist. He is on the editorial board of IJCV, SIIMS, and Computer&Graphics. He has co-founded and served in key executive and technical roles in multiple startup companies, including Novafora, Invision (acquired by Intel in 2012), Videocites, and Fabula AI.

Ron Kimmel is a professor of Computer Science at the Technion where he holds the Montreal Chair in Sciences and a part-time Senior Principal Researcher at Intel RealSense. He held a postdoctoral position at UC Berkeley and a visiting professorship at Stanford University. He has worked in various areas of image and shape analysis in computer vision, image processing, and computer graphics. Prof. Kimmels interest in recent years has been non-rigid shape processing and analysis, medical imaging and computational biometry, numerical optimization of problems with a geometric flavor, and applications of metric geometry and differential geometry. Prof. Kimmel is Fellow of IEEE for his contributions to image processing and non-rigid shape analysis. He is an author of two books, an editor of one, and an author of numerous articles. He is the founder of the Geometric Image Processing Lab at the Technion, and a co-founder and advisor of several successful image processing and analysis companies.