Unsynchronized Structured Light

Daniel Moreno[*] Fatih Calakli[†] Gabriel Taubin[‡]

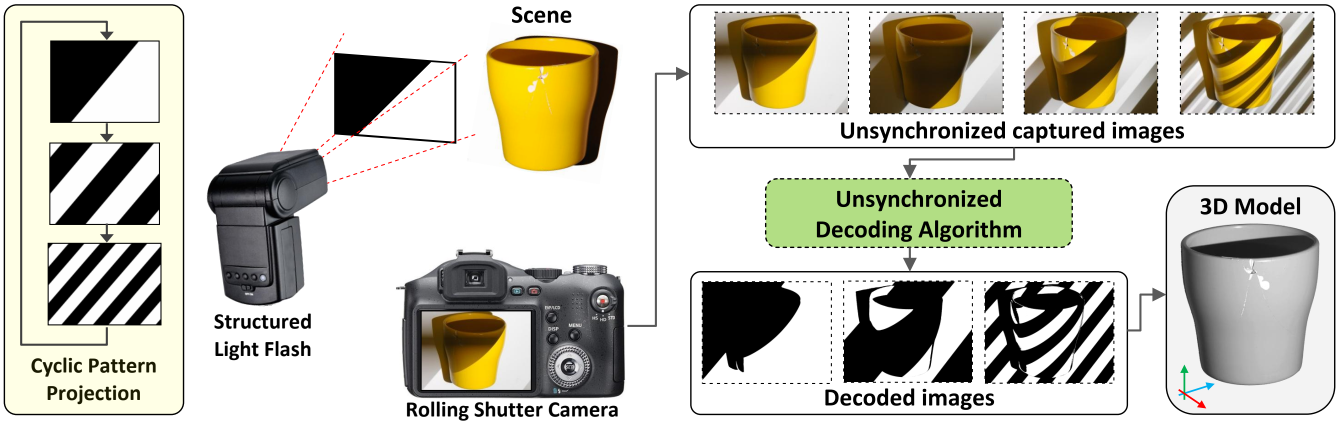

Cyclic Pattern Projection
Structured Light Flash
Scene
Rolling Shutter Camera
Unsynchronized captured images
Unsynchronized Decoding Algorithm
Decoded images
3D Model

Figure 1: *The Unsynchronized Structured Light 3D snapshot pipeline: a) a Structured Light Flash cyclically projects a sequence of* M *patterns onto the scene; b) a Rolling Shutter Camera captures* $N \geq M$ *images without synchronization, which show contributions from consecutive patterns varying row-by-row; c) the Unsychronized Decoding Algorithm decodes the* M *unsynchronized images and generates* N *new images as they would have been captured by a synchronized camera; d) a standard structured light algorithm generates a 3D snapshot, or 3D range image, from the decoded images; and e) multiple 3D snapshots are subsequently registered and a watertight 3D model is reconstructed.*

Abstract

Various Structured Light (SL) methods are used to capture 3D range images, where a number of binary or continuous light patterns are sequentially projected onto a scene of interest, while a digital camera captures images of the illuminated scene. All existing SL methods require the projector and camera to be hardware or software synchronized, with one image captured per projected pattern. A 3D range image is computed from the captured images. The two synchronization methods have disadvantages, which limit the use of SL methods to niche industrial and low quality consumer applications. Unsynchronized Structured Light (USL) is a novel SL method which does not require synchronization of pattern projection and image capture. The light patterns are projected and the images are captured independently, at constant, but possibly different, frame rates. USL synthesizes new binary images as would be decoded from the images captured by a camera synchronized to the projector, reducing the subsequent computation to standard SL. USL works both with global and rolling shutter cameras. USL enables most burst-mode-capable cameras, such as modern smartphones, tablets, DSLRs, and point-and-shoots, to function as high quality 3D snapshot cameras. Beyond the software, which can run in the devices, a separate SL Flash, able to project the sequence of patterns cyclically, during the acquisition time, is needed to enable the functionality.

CR Categories: I.4.1 [IMAGE PROCESSING AND COMPUTER VISION]: Digitization and Image Capture—Scanning

Keywords: 3D, scanning, unsynchronized, structured-light, rolling-shutter

1 Introduction

As the cost and size of digital cameras decreased, the number of them has been increasing dramatically during the last decade. They still exist as dedicated devices such as DSLRs, point-and-shoot, and sport cameras, but their exponential growth has been driven primarily by the use of miniature digital cameras as integral components of every cellphone, tablet, and laptop. Many devices capture uncompressed or mildly compressed high resolution images in burst mode, as well as highly compressed videos at very high frame rates. For example, modern smartphones such as the Apple iPhone 6, and sport cameras such as the GoPro Hero4, are capable of recording 1080p videos at 60fps and 720p videos at 240fps. In parallel, we have seen a growing interest on the application of 3D scanning to a wide range of fields including medicine, forensics, heritage preservation, archaeology, art, metrology, surveillance, and 3D printing.

A typical Structured Light (SL) system projects a sequence of patterns onto a scene while a synchronized camera captures one image of each of them. Synchronization is achieved either by hardware or software means. Hardware synchronization demands specialized high cost devices, and software synchronization results in low

[*] daniel_moreno@brown.edu
[†] fatih_calakli@brown.edu
[‡] gabriel_taubin@brown.edu

ACM Reference Format
Moreno, D., Calakli, F., Taubin, G. 2015. Unsynchronized Structured Light. ACM Trans. Graph. 34, 6, Article 178 (November 2015), 11 pages. DOI = 10.1145/2816795.2818062 http://doi.acm.org/10.1145/2816795.2818062.

SIGGRAPH Asia '15 Technical Paper, November 02 – 05, 2015, Kobe, Japan.
Copyright 2015 ACM 978-1-4503-3931-5/15/11 ... $15.00.
DOI: http://doi.acm.org/10.1145/2816795.2818062

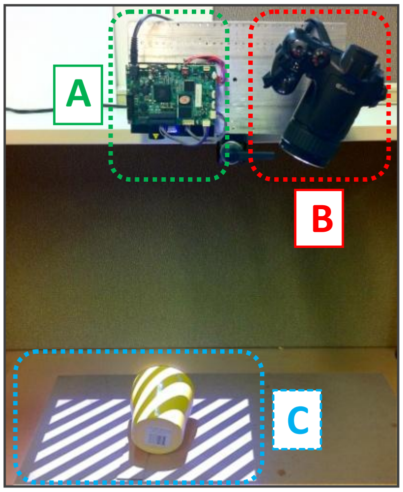


Figure 2: *Acquisition setup: (A) DLP Lightcrafter 4500 projector. (B) Casio Exilim EX-F1 rolling shutter camera. (C) Scene being scanned.*

speed. Neither hardware nor software synchronization works in conjunction with off-the-shelf equipment such as DSLR cameras and smartphones.

In this paper we introduce a novel method called Unsynchronized Structured Light (USL) removing the synchronization requirement altogether. USL enables existing cameras capable of uncompressed or mildly compressed burst-mode image capture (e.g. DSLR's, point-and-shoots, and smartphones) to function as high resolution high precision 3D snapshot cameras only with the addition of an untethered SL Flash, without any hardware modification. USL will be implemented as a firmware or an App running in each device. Figure 1 illustrates a typical USL 3D snapshot pipeline. A standard geometry processing pipeline could follow for registering multiple 3D snapshots and creating watertight 3D models. We believe USL will enable a wide range of applications because it works indistinctly with global and rolling shutter cameras. To the authors' knowledge, USL is the first method that solves the same problem as SL for global and rolling shutter cameras without any kind of synchronization between projection and capture.

The main contributions of the paper are:

- It describes how to implement SL shape measurement in the absence of a hardware or software link between the projection device and the camera.
- It proposes a model for the unsynchronized image formation process in global and rolling shutter cameras, and demonstrates how the model successfully describes images captured with a real rolling shutter consumer camera.
- It demonstrates that the proposed method generates 3D models with no difference in accuracy from those generated with synchronized SL.

2 3D Scanning Technologies

Laser line scanners and time-of-flight (TOF) image sensors are commonly used active shape measurement technologies. Laser line scanners share many characteristics with SL systems, but different from them, their laser light source projects onto the scene a single line (or a small set of lines). Each measurement contains only those scene points illuminated by the laser. They typically include a computer controlled mechanism that incrementally changes the relative position of the lasers with respect to the scene in order to create denser models by aggregating hundreds of individual measurements. The whole process is significantly slower than SL. Differently, TOF image sensors calculate the depth at each pixel measuring the time delay of a modulated light pulse train generated by an active light emitter, usually a laser diode. Measurements are affected by complex noise distributions but they can be generated very fast (e.g. 30 fps) making them suitable for gesture recognition, with Microsoft Kinect 2 being perhaps the best known example for this application. Despite their success on the entertainment business, they currently have very low optical and depth resolution which makes them unsuitable for applications with stricter requirements. SL is the technology of choice for applications requiring highly detailed and precise measurements.

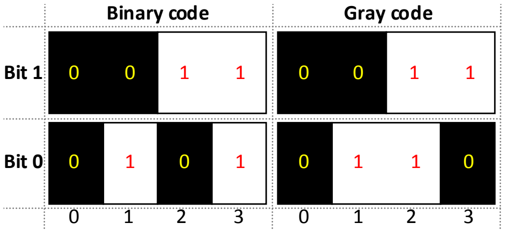


Figure 3: *Binary (LEFT) and Gray (RIGHT) code: 4 projector column indexes are encoded with 2 bits, each bit is represented in a separate pattern as black or white. The numbers at the bottom are the encoded indexes, the numbers within the patterns show the encoded bit values but they are no part of the patterns.*

3 Structured Light

Stereo systems rely on the principles of optical triangulation for measuring the 3D structure of scenes. Two-camera passive stereo systems rely on scene texture and continuity assumptions to solve the correspondence problem. Only scene points for which their projections can be identified in the two cameras result in 3D measurements. SL methods, where one camera is replaced by a digital projector, rely on the projection of a sequence of patterns to solve the stereo correspondence problem without search. The pattern sequence is specially designed to assign codewords to set of pixels. Every coded pixel has its own codeword and a direct mapping exists from codewords to pattern pixel coordinates. The codewords are simply numbers which are represented in the patterns using gray levels, colors, or alternative geometrical representations. Over the years, many SL patterns were created with several categories being proposed to classify them. Salvi et al. [Salvi et al. 2004] classify SL patterns as multiplexed in space or time, and having a discrete or a continuous domain. Time multiplexing encodes codewords along the time axis requiring the projection of several patterns, whereas, spatial multiplexing projects a single pattern with codewords represented as groups of pixels [O'Toole et al. 2014], effectively trading off spatial resolution for number of projected patterns. Discrete pattern codewords uniquely identify discrete positions in the digital projector coordinate space (e.g. integer column indexes). Continuous patterns represent the coordinate space with a continuous property (e.g. the phase of a continuous signal).

Independently of the pattern encoding of choice, existing SL methods assume projector and camera are synchronized so that all pixels, in each captured image, measure the illumination of a single projected pattern. One image is captured per pattern. Single-shot spatially multiplexed methods require no synchronization but they

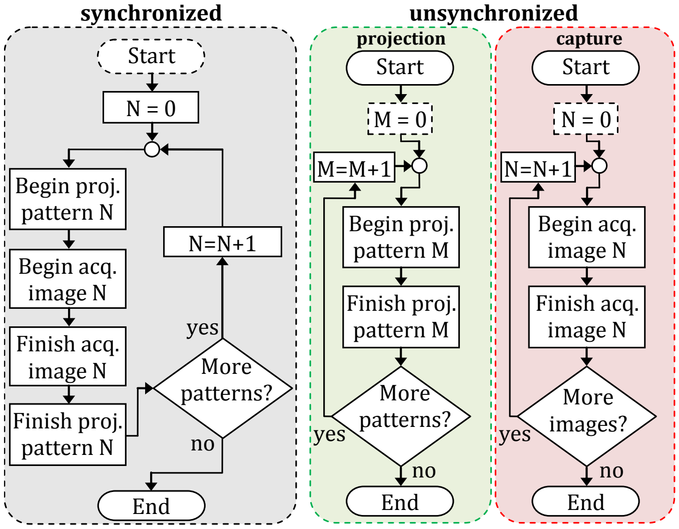


Figure 4: *(LEFT) Synchronized SL: a single control flow must coordinate projection and image acquisition, enforcing the number of images captured to coincide with the length of the pattern sequence. (RIGHT) USL: projection and capture run independently. The number of captured images may differ from the number of projected patterns.*

are not analyzed in this work because of their limited resolution. Synchronized projection and capture has always been assumed to be required, as a result, it has been long ignored in the literature. In practice, precise synchronization is only achieved with hardware triggering, available in custom made systems and expensive industrial equipment. There is no support for hardware synchronization in off-the-shelf consumer products. The alternative is to loosely synchronize the process with a central software controlling all devices. The synchronization software runs on a processor with a connection to all involved devices, such as a central computer where the camera and the digital projector are tethered. Software synchronization performs well only at reduced frame rates because the projection of each pattern has to be paused for a small time to ensure the camera receives and processes each capture command [Jaeggli et al. 2003]. Projection needs to be slowed down even further in systems with rolling shutter cameras, independently of the type of synchronization, to ensure all rows are exposed to the same illumination pattern.

SL systems have a digital projector and at least one camera. We will refer to this minimal case from here on. Our acquisition setup is depicted in Figure 2. Projector and camera are both modeled mathematically with the so-called pinhole model without substantial difference from a typical stereo camera system. For such systems, geometric calibration provides the location, orientation, and other parameters in the model. Ray-ray or ray-plane triangulation of image pixels corresponding to the same scene point result in the 3D coordinate of that point.

4 Binary and Gray code

Binary code uses a sequence of black and white patterns to encode the bits of a set of indexes. Suppose that we want to know which is the column of the pixel in the digital projector illuminating the scene point observed by each camera pixel. Binary code creates a sequence of SL patterns to solve this problem in a simple form. The set of digital projector columns constitutes the integer indexes to encode. Each index is written as a binary number with each bit being assigned to a different pattern. There will be as many patterns as the minimum number of bits necessary to represent all indexes. Each projector pixel is assigned to one index, in our example it is the column number of the pixel. Each pattern corresponds to one position in the binary representation of the indexes. Finally, pattern pixels are painted black if they were assigned to bits with value 0 or painted white otherwise. The example is illustrated in Figure 3 (LEFT) for a fictitious four-column projector.

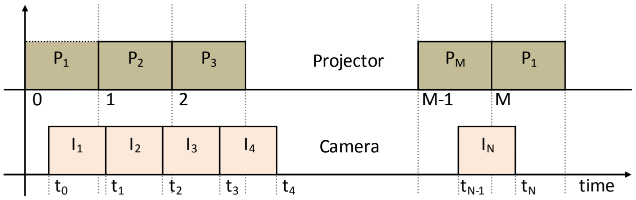


Figure 5: *(TOP) Patterns $P_1 ... P_M$ are cyclically projected at constant framerate. (BOTTOM) An unsychronized camera captures images $I_1 ... I_N$ also at constant framerate. The time unit is identified with one pattern projection time. The values $t_0 ... t_N$ are unknown.*

The SL system projects the binary patterns in sequence and captures an equal number of images with a synchronized camera. Image pixels are analyzed to determine if they were or were not illuminated by the digital projector and translated back to 1 or 0 bit values respectively. In the end, each camera pixel location has associated a bit sequence, one bit per image, representing the index of a projector pixel. In the example we identified indexes with columns because each projector column defines a plane of light. The knowledge of the plane associated to each image pixel enable us to calculate the position of scene points as the intersection of a ray through camera pixels and their associated planes. In summary, the correspondence problem is simplified as the classification of image pixels, either as 0 or 1, based on their intensity values. Binary code is widely used because of its simplicity and robustness to varying illumination levels and light color.

Camera pixels imaging points in a pattern transition region, either from black to white or the opposite, have intermediate values and the probability of making a mistake while classifying them increases substantially. For such reason, it is common to encode the set of indexes with Gray code prior to creating the binary patterns, idea first proposed in [Inokuchi et al. 1984]. Figure 3 (RIGHT) shows the indexes in the example encoded with Binary Reflected Gray Code (BRGC); observe how there is one less transition in the Bit 0 pattern than in its Binary code version. The reduction in the number of transitions diminishes the number of decoding errors.

Projected light is sometimes reflected on the scene to other scene points before reaching the camera sensor. These interreflections may cause additional decoding errors because they alter the intensities measured at the camera. Interreflections are dependent on the scene geometry and materials. Nayar et al. proposed a simple method, [Nayar et al. 2006], to split captured images in a direct component having only direct projector light contributions, and a global component with contributions from interreflections and other non-direct effects. The explicit separation of the two components may reduce decoding errors because it allows the decoding algorithm to make better classification decisions [Xu and Aliaga 2007]. An alternative and complementary method to reduce these errors is to design projection sequences with patterns having all similar spatial frequencies. Similar pattern frequencies result in similar global components affecting all pixels. Based on this observation, Gupta et al. projected Gray code patterns with vertical stripes all of similar width [Gupta et al. 2011], contrary to BRGC. Same as us, they used

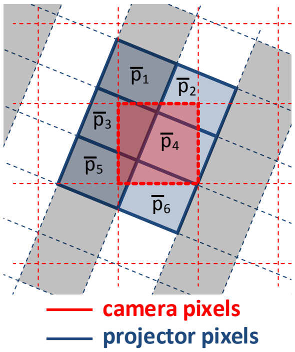


Figure 6: *Image formation: the red grid represents the pixels at the camera sensor,* p *is the central pixel; the blue grid corresponds to the data projector pixels illuminating the scene as seen from the camera viewpoint. The figure illustrates that only a subset of all data projector pixels will contribute to the intensity at each camera pixel. In this figure, only* $\bar{p}_1$*, ...,* $\bar{p}_6$ *will contribute to* p*.*

a Gray code sequence with minimum stripe width of 8 pixels called MinSW8, first proposed in [Goddyn and Gvozdjak 2003]. Moreno et al. applied a similar idea to phase shifting [Moreno et al. 2015].

5 Related Work

Active illumination methods to capture 3D data are not new. Shirai and Suwa [Shirai and Suwa 1971] proposed the first slit line scanner. Posdamer and Altschuler were the first to use binary patterns for 3D scanning [Posdamer and Altschuler 1982]. Inokuchi et al. made the method more robust to noise by using Gray code instead [Inokuchi et al. 1984; Sato 1987]. Srinivasan et al. replaced the discrete patterns with continuous phase shifting, encoding one dimension in the projector coordinate space as the phase of a sinusoidal signal [Srinivasan et al. 1985]. These works are not only important for their historical impact. They remain relevant today as the most used methods for 3D shape measurement: the slit line scanner is the working principle of many industrial laser scanners, Gray code is the most widely used discrete pattern in structured light, and the same goes for phase shifting within the continuous domain. Salvi et al. [Salvi et al. 2004] have written an excellent survey on SL codification strategies. SL methods based on continuous patterns are surveyed by several authors [Salvi et al. 2010; Zhang 2010; Bell and Zhang 2014].

Koppal et al. used an unsynchronized data projector with a high speed camera for reconstruction and photography of high-speed scenes [Koppal et al. 2012]. We find theirs the closer work to ours, however, it is substantially different. They have used a camera operating at 3000 fps to create a fingerprint of the data projector dithering pattern for different projected intensities and used the fingerprints with a look-up table (LUT) to solve the correspondence problem and other applications. On the contrary, our method uses a camera at the same or slightly higher frame rate than the data projector and rather than searching in a LUT we estimate the projected values by optimizing the parameters of the unsynchronized image formation model we have created. Fujiyoshi et al. created a stereo system with two unsynchronized cameras which uses the most recent image in any of the cameras together with a Kalman filter to predict the 3D position of a moving object [Fujiyoshi et al. 2003]. Hasler et al. used unsynchronized cameras to capture the motion of a person, their method synchronizes the video streams offline by analyzing their audio channels [Hasler et al. 2009]. Bradley et al. studied how to synchronize an array of rolling shutter cameras [Bradley et al. 2009], they proposed to use either stroboscopic illumination or optical flow. Finally, Hu et al. implemented a communication system between an LCD display and a rolling shutter camera [Hu et al. 2013] which has some resemblance to SL if the projector and camera is thought as a communication channel. Previous systems, e.g. [Langlotz and Bimber 2007], required the camera to run at twice the display frame rate as demanded by the Nyquist-Shannon limit in sampling theory. Hu et al. improved on those methods by modifying the displayed patterns, although, their work is not applicable to SL where the goal is to decode each individual pixel rather than a complete pattern.

6 Unsynchronized Structured Light

USL differs from the standard version in that the synchronization assumption is removed. The immediate consequence is that without synchronization the digital projector may switch patterns at any moment in the image capture timeline. It will often happen that a switch occurs during the exposure of some camera pixels, in which case, the camera will output pixel intensities that are a combination of the projected values of more than one pattern. Global shutter cameras expose all their pixels simultaneously; if used for Unsychronized SL, the amount of contribution from individual patterns will be identical for every pixel in each image. On the other hand, rolling shutter cameras, which expose rows one-by-one, will have varying pattern contributions within a single image. Observe in Figure 1 how pattern intensities vary from top to bottom in the unsychronized images: one pattern becomes more visible while the other vanishes. USL requires that projector and camera operate at a constant, possibly different, frame rate. Unequal frame rates may result in a difference on the number of projected patterns and captured images; another distinction from SL where the two numbers always match. Figure 4 compares the acquisition control flow in the synchronized and the unsynchronized methods; the last one having independent projection and capture modules naturally allows capturing more images than projected patterns, a situation illustrated in Figure 1 where the projected sequence has only three patterns but four images were captured.

The complete pipeline of USL is depicted in Figure 1. A sequence of patterns is projected onto the scene continuously by a SL Flash with no connection whatsoever to the capture hardware and software. A camera captures a sequence of unsynchronized images long enough to cover all projected patterns, each image may have contributions of different patterns. The image set is processed by the Unsychronized Decoding Algorithm which generates a new sequence of images, with length equal to the pattern sequence. The newly created binary images are equivalent to those from SL and can be used in combination with existing algorithms to generate a 3D model of the scene.

6.1 Problem Formulation

A sequence of M black and white patterns, P_1,...,P_M, is cyclically projected onto a scene at constant frame rate. Simultaneously, a camera captures $N \geq M$ images of the scene, I_1,...,I_N, at constant frame rate. Camera frame rate is equal or greater than projection frame rate, and N is large enough to observe the M patterns at least once. The capture of I_1 begins while P_1 is being projected and the capture of I_N while P_M is being projected, see Figure 5. The Unsynchronized Decoding Algorithm creates a new sequence of M images in correspondence to the projected patterns as if they were captured by a synchronized camera, that is, given an unsynchronized image sequence it synthesizes a synchronized image sequence (Figure 1). The newly created sequence is the output of the algorithm and it could be processed with other SL algorithms.

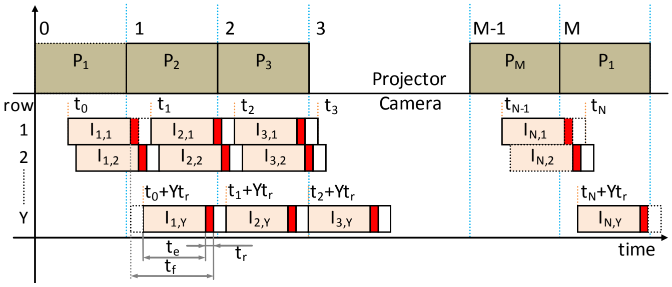


Figure 7: *(TOP) Patterns $P_1 ... P_M$ are cyclically projected at constant framerate. (BOTTOM) Image rows captured by a rolling shutter camera: $I_{n,y}$ corresponds to row y in image n, t_e and t_r are the exposure and readout time for a single row, and t_f the time to read all pixels in a single image out the camera sensor. Row indexes begin at 1.*

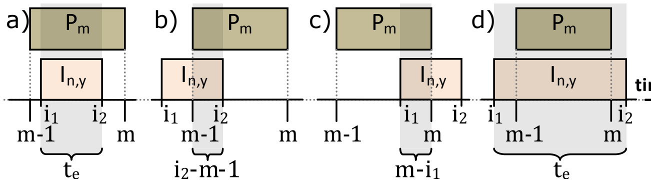


Figure 8: *Conditions on definition of v_{nmy} in Equation 17: a),b),c) are the only 3 cases where P_m contributes to $I_{n,y}$; d) never occurs because camera framerate is higher than data projector's.*

We set the time origin, $t=0$, to the instant when projection of P_1 begins, and we identify the interval between the projection beginning of two consecutive patterns with the time unit, i.e. P_m is projected from time $m-1$ to m. Similarly, we call t_{n-1} the unknown time at which the capture of I_n begins.

6.2 Image Formation

Camera sensors are designed to measure the amount of light received at each pixel element during the exposure time. These measurements are translated by the camera software into intensity values and stored in an image. Here, we provide a model explaining the intensity values measured by the camera in a SL system. The model corresponds to a single pixel, therefore, it is valid for global and rolling shutter cameras. Let us consider the intensity measured at a camera pixel p, exposed from time t_0 to t_1, as expressed by Equation 1. The total measured intensity I_p results from integrating over time the contribution from all data projector pixels $\bar{p}$. The function $P_{\bar{p}}(t) \in [0,1]$ represents the value of data projector pixel $\bar{p}$ at time t. The scalar $R(p,\bar{p}) \in [0,1]$ measures the contribution of $\bar{p}$ to the camera pixel p, which is non-zero only for the small set of data projector pixels illuminating the scene patch imaged by p. In the example of Figure 6, only $R(\bar{p}_1,p)$, $R(\bar{p}_2,p)$, ..., $R(\bar{p}_6,p)$ are non-zero. S_p is a constant relating projector and camera intensity levels, and $C_p(t)$ is the global illumination term representing all other light contributions received at the scene point not directly coming from the digital projector.

$$I_p = \int_{t_0}^{t_1} S_p \sum_{\bar{p}} R(p,\bar{p})\, P_{\bar{p}}(t)\, dt + \int_{t_0}^{t_1} C_p(t)\, dt \tag{1}$$

We assume that $\int C_p(t)\, dt = Cp$ is constant to make the problem tractable. We are interested in binary patterns only, thus, we are restricting $P_{\bar{p}}(t)$ to values $\{0,1\}$, and we will assume that digital projectors toggle pixels instantaneously. Furthermore, we assume that a single projector pixel is responsible for most of the direct illumination in each camera pixel (e.g. $\bar{p}_4$ in Figure 6). We will make the approximation of considering this main contribution as the *only* direct contribution to each camera pixel. We have found the approximation correct for our setup where the camera has larger resolution than the digital projector, moreover, pixels in Binary and Gray code patterns share identical values with several of their neighbors. In Figure 6, $\bar{p}_4$ has same value as neighbors $\bar{p}_2$ and $\bar{p}_6$. The image formation model, taking all this considerations into account, results in the simpler Equation 2. Observe that, for structured light, the only known value in the model is the intensity Ip measured by the camera. The SL problem is to determine which projector pixel $\bar{p}$ contributed to p. USL must first recover the function $P_{\bar{p}}(t)$, and then proceed to solve the SL problem.

$$I_p = \int_{t_0}^{t_1} S_p\, R(p,\bar{p})\, P_{\bar{p}}(t)\, dt + C_p \tag{2}$$

6.3 Image Normalization

The measured intensity I_p, in the simplified image formation from Equation 2, depends on several unknown variables in addition to the value of the projected pattern. In USL we normalize all captured images to avoid working with these additional variables which are of no interest for our problem. Let be I_p^1 the value of Equation 2 when $P_{\bar{p}}(t) = 1$ and I_p^0 its value when $P_{\bar{p}}(t) = 0$. We define a normalized image $\tilde{I}_p$ as follows

$$\tilde{I}_p = \frac{I_p - I_p^0}{I_p^1 - I_p^0} = \frac{1}{t_1 - t_0} \int_{t_0}^{t_1} P_{\bar{p}}(t)\, dt. \tag{3}$$

The normalized image formation equation only depends on the integration time and the value of the projected pattern. A caveat is that Equation 3 assumes that C_p is constant for all images which may not always be true (e.g. interreflections vary with different patterns). The following sections only make use of normalized images and no further distinction in the notation is made.

6.4 Global Shutter Unsynchronized Camera

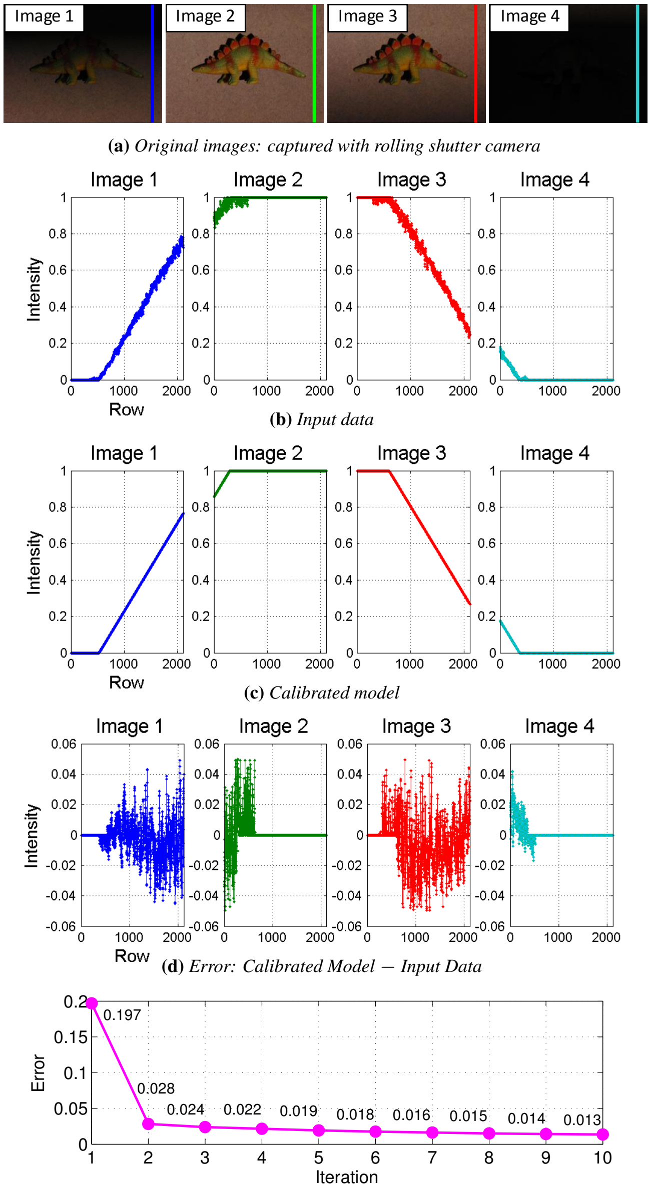
In the previous section, the normalized image formation equation describes the measured intensity at a camera pixel p for a single image. We now proceed to find the value of p in all the images in the unsynchronized sequence for a global shutter camera. In this section, we refer to a single camera and a single data projector pixel; we also drop the subscripts p and $\bar{p}$ to simplify the notation, i.e. I_n is the intensity at pixel p in the normalized image n and P_m is the projected value at pixel $\bar{p}$ in pattern m. Let be $t_e = t_n - t_{n-1}$ the unknown image exposure time. The exposure of I_n begins at time $t_n = t_0 + n\, t_e$. We now rewrite Equation 3 as

$$I_n = \frac{1}{t_e} \int_{t_{n-1}}^{t_n} \sum_{m=1}^{M} r_m(t) P_m\, dt = \frac{1}{t_e} \sum_{m=1}^{M} P_m \int_{t_{n-1}}^{t_n} r_m(t)\, dt \tag{4}$$

$$r_m(t) = \begin{cases} 1 & \text{if } m-1 \le t \le m \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

In Equation 4, I_n depends on the fixed exposure time t_e, the M pattern values of $\bar{p}$, and a selector function $r_m(t)$. We group all projected values in a vector $P = (P_1, \ldots, P_M)^T$. We define a new function $\phi_n : \mathbb{R}^2 \rightarrow \mathbb{R}^M$ that makes the following equation true

$$I_n = \phi_n(t_e, t_0)^T P, \tag{6}$$



(a) *Original images: captured with rolling shutter camera*

(b) *Input data*

(c) *Calibrated model*

(d) *Error: Calibrated Model — Input Data*

(e) *Optimization convergence: Error vs. Iteration number*

Figure 9: *Time variables calibration. (9a) Original images: color lines highlight columns used for calibration. (9b) Input data: plot shows pixel intensities along a single column of four consecutive normalized images captured with a rolling shutter camera while projector switches from black to white and to black again. (9c) Model data: pixel intensities predicted by the calibrated model. (9d) Error: difference between calibrated model and input data pixel-by-pixel. Intensity values predicted by calibrated model differ less than 0.05 from measured data. (9e) Optimization error: RMSE at each of the iteration end.*

a suitable definition for ϕ_n is

$$\phi_n(t_e,t_0) = (\phi_{n,1}(t_e,t_0),\dots,\phi_{n,M}(t_e,t_0))^T, \tag{7}$$

with each component $\phi_{n,m}$ given by

$$\phi_{n,m}(t_e,t_0) = \frac{1}{t_e}\max\big(0,\min(m,t_n)-\max(m–1,t_{n-1})\big). \tag{8}$$

It is clear from Equation 8 that ϕ_n is not a linear function of t_e and t_0. The known value of I_n depends only on the unknown time t_0 at which begins the image sequence capture relative to P_1, the fixed exposure time t_e, and the pattern values P_1,...,P_M unknown. The fact that the projected pattern sequence is chosen by us and known must not mislead the reader to think that P_1,...,P_M are also known. The value of all pattern pixels is known but it is unknown which projector pixel $\bar{p}$ illuminates the scene point imaged by the camera at p. Identifying these projector-camera pixel correspondences is the standard SL problem to be solved. Finally, we group the image intensities at p in a vector $I = (I_1,\dots,I_N)^T$ to write

$$I = \Phi(t_e,t_0)^T\,P, \tag{9}$$

where Φ is a measurement matrix with column n equal to $\phi_n(t_e,t_0)$. Φ is the same for all pixels in the sequence and it is completely determined by t_e and t_0. In the special case where camera and projection frame rates are equal, Φ is square and bidiagonal. Moreover, if the frame rates are equal and $t_0 = 0$, then Φ is the identity matrix as in SL. In general, Φ is an $M\times N$ rectangular matrix and $\Phi\Phi^T$ is tridiagonal with non-zero diagonal elements and non-negative elements. Thus, the system $\Phi^T P = I$ has a unique solution.

6.5 Rolling Shutter Unsynchronized Camera

Rolling shutter cameras are omnipresent today. Different from global shutter cameras, they expose and read out rows sequentially from the camera sensor and compose all rows into an output image. The result is an image where each row was captured at a slightly later time than previous rows within the image. In addition, they begin the exposure of rows in the next image before finishing the capture of the current image. We need to include these new time relations in the image formation equation in order to perform unsynchronized captures with this type of cameras (Figure 7).

The time t_e is now the exposure time of a single row. We have added the readout time t_r, which is the time to transfer one row from the image sensor to the camera internal memory, and t_f, the time required to capture a complete image. The maximum frame rate of any rolling shutter sensor is the reciprocal of the time required to read out all sensor rows: $\textbf{sensorRows} \times t_r$. Cameras usually crop the data read from the image sensor as part of their post-processing algorithms, as result, the number of rows Y in the images may be smaller than the total sensor rows making $t_f \neq Y\,t_r$, although, the relation $t_f \geq Y\,t_r$ is always true. Another useful relation is $t_e + t_r \leq t_f$. We refer to row y in image n as $I_{n,y}$. Its exposure begins at time $t_{n,y}$. We define $t_n = t_{n,1}$ as shorthand.

$$t_n = t_0 + n\,t_f,\quad t_{n,y} = t_n + (y-1)\,t_r. \tag{10}$$

The simplified row image formation equation is

$$I_{n,y} = \frac{1}{t_e}\sum_{m=1}^{M} P_m \int_{t_{n,y}}^{t_{n,y}+t_e} r_m(t)\,dt, \tag{11}$$

which we rewrite as

$$I_{n,y} = \phi_{n,y}(t_e,t_f,t_r,t_0)^T P, \tag{12}$$

using the following definitions:

$$\phi_{n,y}(t_e,t_f,t_r,t_0) = \begin{bmatrix}\phi_{n,1,y}(t_e,t_f,t_r,t_0)\\ \vdots \\ \phi_{n,M,y}(t_e,t_f,t_r,t_0)\end{bmatrix}, \tag{13}$$

$$\phi_{n,m,y}(\cdots) = \frac{1}{t_e}\max\big(0,\min(m,t_{n,y}+t_e)-\max(m–1,t_{n,y})\big). \tag{14}$$

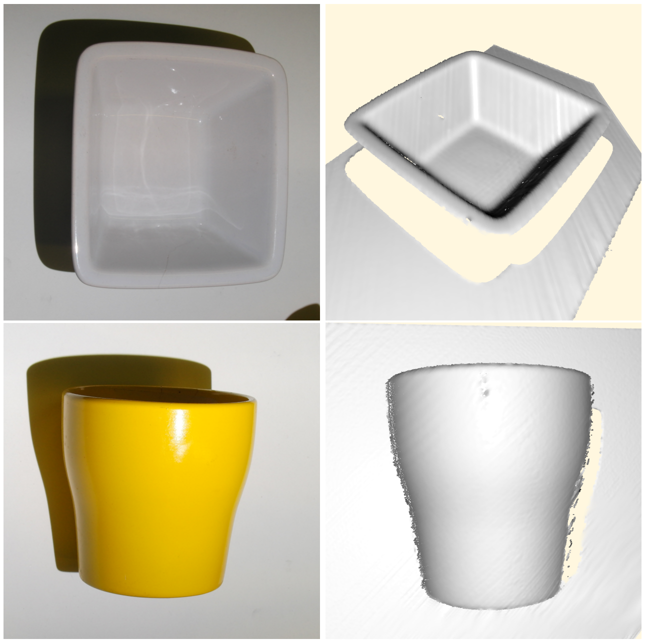


Figure 10: *Example 3D models: (LEFT) picture of the scene, (RIGHT) 3D models created with USL. Models have no visible difference with those created with SL.*

Similar as in Section 6.4, it is clear from Equation 14 that $\phi_{n,y}$ is not a linear function of its parameters. Collecting vectors $\phi_{1,y}$,...,$\phi_{N,y}$ in a measurement matrix Φ_y allow us to write

$$I_y(x,y) = \Phi_y(t_e,t_f,t_r,t_0)^T P(x,y), \qquad (15)$$

which gives the relation between the sequence of projected patterns P and the sequence of observed image intensities at row y. Φ_y depends only on the variables t_e, t_f, t_r, and t_0, and it is a fixed matrix for all pixels at row y in the image sequence. The value of t_e, t_f, and t_r are intrinsic parameters of the camera, whereas, t_0 varies from sequence to sequence.

7 The Unsynchronized Decoding Algorithm

We have presented image formation models for global and rolling shutter cameras. We now present a decoding algorithm which estimates all the unknown variables in those models. Without loss of generality, we present only the algorithm for rolling shutter cameras. Solving USL for rolling shutter cameras is more challenging and it has many applications given the large number of these cameras readily available for consumers. Moreover, this model can be applied to global shutter cameras by making $t_r = 0$. All the results that follow are equally valid for rolling and global shutter cameras.

The Unsynchronized Decoding Algorithm estimates P (the projected pattern values) for each camera pixel observing the target scene. Equation 15 models the relation between P and I_y, the unknowns are t_e, t_f, t_r, t_0, and P. We proceed in two steps:

1. **Calibration of Time Variables**: solve for variables t_e, t_f, t_r, and t_0 which define the measurement matrix Φ_y.

2. **Estimation of Pattern Values**: find values for P agreeing with matrix Φ_y and observed image intensities.

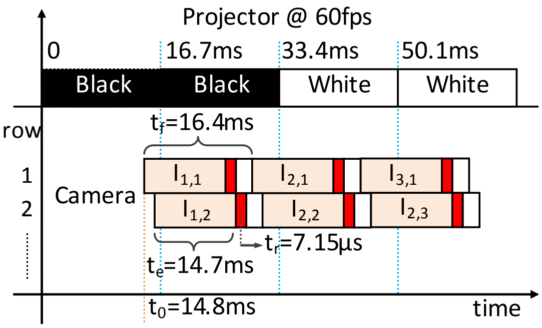


Figure 11: *Calibrated time variables: final estimated values for t_e, t_f, t_r, and t_0 in the calibration example.*

7.1 Calibration of Time Variables

We consider t_e, t_f, and t_r intrinsic camera parameters which could be known from the camera specification or because they were previously calibrated. On the contrary, t_0 has to be estimated for every single sequence. In practice, if they are all unknown, it is useful to estimate them simultaneously which is the path we follow here. Equation 15 gives the relation between the observed image intensities I_y, the projected patterns P, and the measurement matrix Φ_y, but it gives no information if both P and Φ_y are unknown. We break this dilemma by adding to the pattern sequence a few additional patterns, specifically, we prepend the subsequence {B,B,W,W,B} to all projected sequences where B and W are patterns with all black and all white pixels respectively. The knowledge provided by the extra patterns permits to write a reduced version of Equation 15 considering only the first four projected patterns, $P = (0,0,1,1)^T$, and corresponding images, with Φ_y the only unknown. Pixels within these patterns all have identical values.

Let us define $h = \frac{1}{t_e}(1,t_f,t_r,t_0)^T$ and write each scalar $\phi_{n,m,y}$ in Φ_y as the following dot product:

$$\phi_{n,m,y}(h) = v_{nmy}(h) \cdot h \qquad (16)$$

with the vector v_{nmy} defined based on the overlap between the exposure of $I_{n,y}$ and the projection of P_m. The exposure of $I_{n,y}$ begins at $i_1 = t_0 + n\,t_f + (y-1)\,t_r$ and ends at $i_2 = i_1 + t_e$. A continuous nonlinear definition of v_{nmy} is

$$v_{nmy}^T(h) = \begin{cases} [\quad t_e, \quad 0, \quad 0, \quad 0] & \text{if } m-1 < i_1 < i_2 < m \\ [t_e - m+1, n-1, \quad y, \quad 1] & \text{if } i_1 \le m-1 < i_2 < m \\ [\quad m, 1-n, -y, -1] & \text{if } m-1 < i_1 < m \le i_2 \\ [\quad 1, \quad 0, \quad 0, \quad 0] & \text{if } i_1 \le m-1 < m \le i_2 \\ [\quad 0, \quad 0, \quad 0, \quad 0] & \text{otherwise} \end{cases} \qquad (17)$$

Recall that $\phi_{n,m,y}$'s are the components of the vector $\phi_{n,y}$ which, with the above definition, can be written as

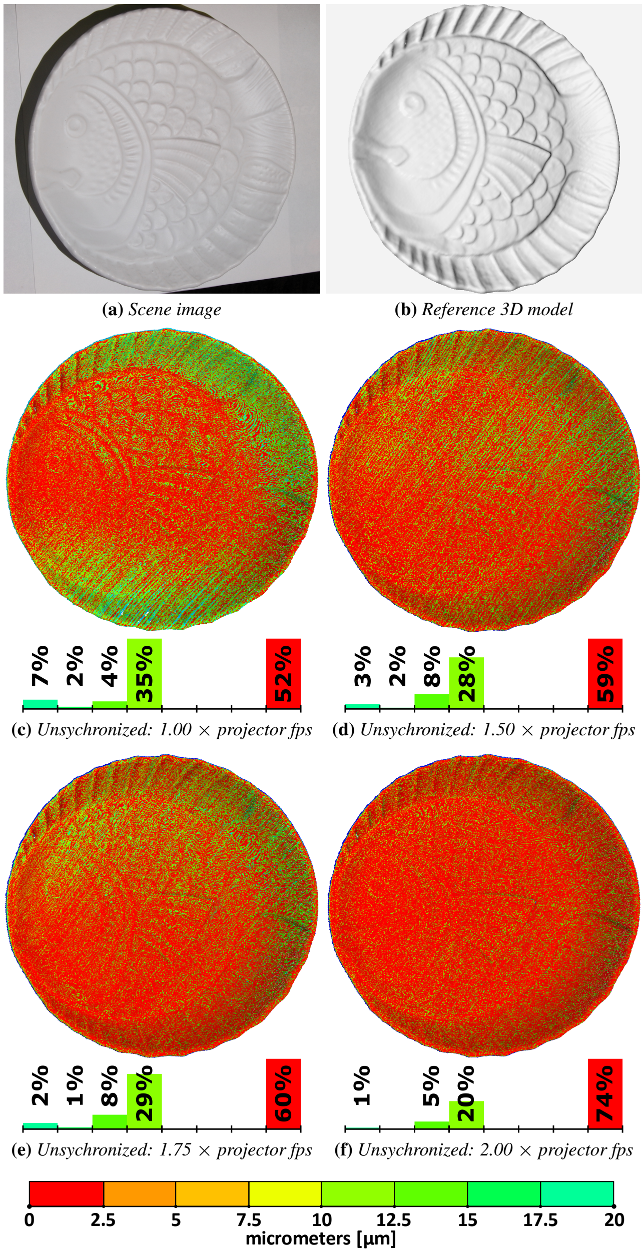
$$\phi_{n,y} = (v_{n1y}^T h, \ldots, v_{nMy}^T h)^T \overset{\text{def}}{=} V_{ny} h. \qquad (18)$$

We use Equation 18 to write the following nonlinear energy function

$$E(h) = \frac{1}{2} \sum_n \sum_{x,y} \left(P(x,y)^T V_{ny} h - I_n(x,y) \right)^2, \qquad (19)$$

which we minimize, in respect to h, with the following constraints

$$h > \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \text{ and } \begin{bmatrix} 0 & -1 & Y & 0 \\ 0 & -1 & 1 & 0 \end{bmatrix} h \le \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \qquad (20)$$



(a) *Scene image* **(b)** *Reference 3D model*

(c) *Unsychronized: 1.00 × projector fps* **(d)** *Unsychronized: 1.50 × projector fps*

(e) *Unsychronized: 1.75 × projector fps* **(f)** *Unsychronized: 2.00 × projector fps*

Figure 12: *Acquisition frame rate: 3D models were created with an unsychronized rolling shutter camera running at 1, 1.5, 1.75, and 2 times the projecton frame rate. A synchronized camera was used to create a reference 3D model (12b). The distance between each model and the reference is shown color coded in (12c), (12d), (12e), and (12f), with a histogram of the distances below each figure. The color scale is displayed at the bottom. (12a) is a picture of the scene, a ceramic plate of 20cm diameter. Every time 15 patterns were projected and 15 (12c), 23 (12d), 26 (12e), and 30 (12f) images were captured. A pixel in the data projector corresponds to about 30µm on the scene surface, for all models the majority of the points coincide exactly with those measured with Standard SL, the rest differ on about half the size of a pixel.*

which are equivalent to: $t_e > 0$, $t_r > 0$, $t_0 > 0$, $t_f > t_e$, $Y t_r \leq t_f$ and $t_r + t_e \leq t_f$. We minimize Equation 19 iteratively. At each iteration, the current $h^{(k)}$ gives V_{ny} and we use the nonlinear least squares solver in MATLAB to find the next value $h^{(k+1)}$. The energy is not linear with several local minimums, however, we found experimentally that with proper initial values it converges very fast to the desired solution. We used $t_e = 1$, $t_f = 1.1 t_e$, $t_r = 0.0001$, and $t_0 = 0.5 t_e$ as initial values in all cases except that we set $t_e = 2$ when the camera frame rate was significantly increased. The optimized values of t_e, t_f, t_r, and t_0 extracted from h are used with Equation 13 to obtain Φ_y.

7.2 Estimation of Pattern Values

The estimation of the sequence of projected pattern values $P(x, y)$ at camera pixel (x, y) is independent of all other camera pixels and it could be run in parallel with a multithread implementation or in the graphics hardware. In this section we describe the algorithm for a single pixel. Since we work with binary patterns, $P_m(x, y) \in \{0, 1\}$, solving the optimal P in Equation 15 requires integer programming which is NP-hard. Instead, we relax the problem, enforcing only box constraints on P, and we use energy minimization to find a suitable solution which is later binarized to get a valid $P(x, y)$. We propose the following energy function

$$\begin{aligned} E(P) = \frac{1}{2} \sum_{n=1}^{N} \left(h^T V_{ny}^T P(x,y) - I_n(x,y) \right)^2 \\ + \frac{\lambda}{2} \sum_{m=1}^{M} P_m(x,y) \left(1 - P_m(x,y)\right) \end{aligned} \qquad (21)$$

which is quadratic on P and has a global minimum. The second term is an inverted parabola favoring $\{0, 1\}$ values of P_m chosen as regularization. The optimization problem to solve is

$$\hat{P}(x,y) = \underset{P}{\operatorname{argmin}}\, E(P) \;\; \text{s.t.}\; 0 \leq P_m \leq 1,\; m{:}1 \ldots M. \qquad (22)$$

We minimize Equation 22 with our own implementation of a box-constrained conjugate gradient algorithm. We choose the initial value of P randomly. Because we work with normalized images, we binarize the solution with a threshold at the middle value 0.5.

8 Comparison with alternative methods

The problem of creating a SL sequence for 3D scanning without synchronization could also be solved by increasing the camera frame rate enough to guarantee there is one image observing a single pattern for every projected pattern. For instance, a global shutter camera running at twice the projector frame rate would capture one complete pattern every two frames. This method would allow to create a SL sequence of 12 patterns (10 Gray code plus one Black and one White pattern for image normalization) in the time necessary to capture 24 images, although, it requires exact calibration of camera and projector frame rates. If the exact relation between frame rates is unknown then an algorithm is required for scanning the captured sequence and selecting those images observing a single projected pattern. The implementation of such algorithm for an uncalibrated rolling shutter camera is somewhat more complex because in addition to identifying which rows observe a single pattern, the algorithm would have to identify which pattern each row is observing at and synthesize synchronized images by picking rows from different images ensuring they all belong to the same pattern. We have not implemented such algorithm because we are interested in running the camera and projector at approximately the same frame rate to reduce the image acquisition time. Nevertheless, it was suggested to us that the alternative algorithm may be simpler to implement because it would skip the optimization step performed by USL. At this time we have no proof supporting this

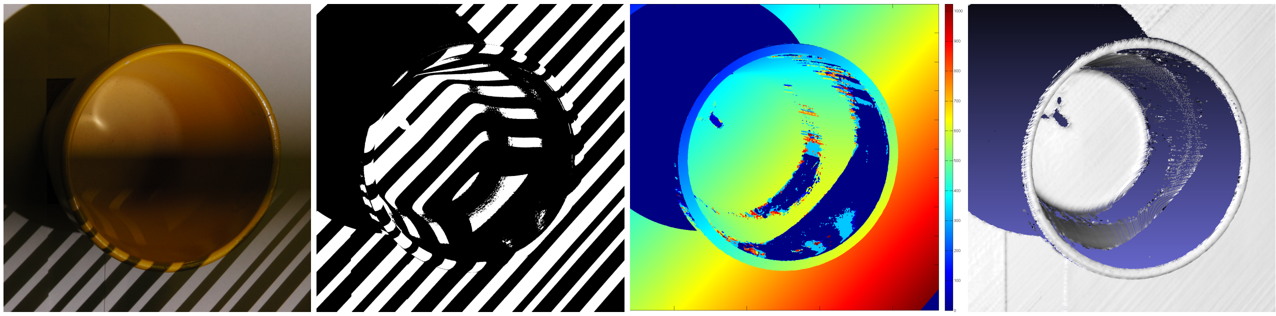


Figure 13: *Influence of global illumination effects. USL applied to the interior of a deep concave cup. From left to right: one image in the input sequence captured during a pattern change; one binary pattern generated by USL; color coded image of the result of SL, each color corresponds to a projector diagonal line; and a render of the final 3D model. Interreflections on the cup interior caused that the right wall could be only partially measured. The same scene scanned with synchronized SL produces identical results.*

claim neither the opposite. If there were no constrains on the acquisition time we would have projected patterns encoding rows and columns and their complementary patterns for improving accuracy and robustness, quadruplicating the acquisition time, as it is a common practice in standard SL. Instead, we have chosen to solve the more complex USL problem as presented in order to minimize the acquisition time. In addition, we are currently working on alternatives for the time variables calibration without adding the extra patterns and reducing further more the acquisition time.

9 Results

Our acquisition setup is depicted in Figure 2. It consists of a Casio Exilim EX-F1 camera and a DLP LightCrafter 4500 data projector. The camera features a burst mode that captures up to 60 full frame images in 1 second, which are stored in compressed JPG format with a resolution of 2816×2112 pixels. We consider this particular camera an upscale point-and-shoot because it lacks of features available in other DSLR's like optical viewfinder, optical zoom control, access to burst mode raw image data, and interchangeable lenses. We tried other cameras claiming to capture HD video as the Olympus Stylus VH-520 only to find that they create h.264 videos with a level of compression so high that it looses most high frequencies and that makes frames extremely 'blocky'. They also apply variable frame rate compression, not fitting with the assumption of our method. We believe that manufacturers could modify cameras like that to feed the image data to the Unsynchronized Decoding Algorithm, instead of to the h.264 compression module, to enable them to capture 3D snapshots using our method. Meanwhile, we have chosen to run the experiments with the Casio Exilim EX-F1 to prove the feasibility of USL in cameras with a rolling shutter imaging sensor. We built a prototype of the SL Flash with the TI DLP LightCrafter 4500 development kit programmed to cyclically project the sequence of patterns whenever it is turn on. The device has a native resolution of 912×1140 and it supports up to 4225Hz binary pattern projection frame rate. In all our experiments, we have projected a sequence of 15 patterns consisting of the 5 patterns used for the time variable calibration followed by 10 bits of the MinSW8 Gray code, with no separation or marker to divide the sequence components or a sequence cycle from the next one. Our device simply projects the 15 patterns in a fixed order over and over. Our Gray code pattern indexes represent projection diagonals, instead of encoding columns, because the DLP sensor in our device has the so-called 'diamond pixel configuration', meaning that pixel elements are rotated 45 degrees. By rotating the projection patterns in identical fashion, we have aligned their transition regions with the pixel divisions in the projection sensor. The images I_p^0 and I_p^1 used for image normalization were obtained as the minimum and maximum of the unsynchronized image sequence. Finally, the setup was geometrically calibrated with the Projector-Camera Calibration Tool [Moreno and Taubin 2012].

9.1 Time Variables Calibration results

Figure 9 shows typical input data and result of the time variables calibration, described in Section 7.1, of a rolling shutter camera running at approximately the same frame rate as the digital projector. Because of the similar frame rates, 4 images, Figure 9a, were enough to capture the initial subsequence of two black and two white patterns used for calibration. The color lines highlighted correspond to the image columns used for calibration. The calibration subsequence was manually selected from the whole image sequence. Figure 9b shows normalized intensities at the highlighted image columns. Observe that the projection switches from black to white while the last rows in Image 1 and the first rows in Image 2 are being exposed, and then it switches from white to black while recording Image 3 and 4. Figure 9c shows the intensity values predicted by the calibrated model using Equation 15, after minimizing Equation 19 for 10 iterations. Figure 9d is the difference between (9c) and (9b) showing that the final error is low; the small variation, less that 0.05, corresponds to noise in the input data. Figure 9e is the root mean square error (RMSE) during optimization which converged in a few iterations.

$$RMSE = \sqrt{\frac{1}{N}\sum_{x,y,n}(\hat{I}_n(x,y) - I_n(x,y))^2} \qquad (23)$$

The actual result of the calibration are the estimated values for t_e, t_f, t_r, and t_0. Figure 11 gives their value for this particular example. The presented case corresponds to a single example which the authors believe represents well the calibration results seen during all experiments. In cases where the camera runs at a higher frame rate, the number of patterns is kept fixed but extra images are used for calibration, e.g. 8 images are required to observe all 4 patterns with a camera running at twice the projection frame rate.

9.2 Estimation of Pattern Values results

Results for the pattern value estimation algorithm from Section 7.2 are displayed in Figure 14. On its left are the images as read from the camera memory, whereas, on its right are the binary images created by the Unsynchronized Decoding Algorithm. Camera and projector were both running at 60 fps in this example. The equal frame rate case is the most challenging one because every image has contributions from two patterns and each pattern is only imaged in two images, except in the unlikely case that $t_0 = 0$ corresponding to the synchronized case. Captured images as these ones could not be correctly binarized with the state-of-the-art SL techniques. Our algorithm succeeds because it makes use of information from all the images in the sequence to create a novel synchronized image for each projected pattern. Unlike images from rolling shutter cameras, all rows in every output image are a capture of the same single pattern as in the synchronized case.

9.3 Camera and projection frame rate ratio

USL allows to choose the ratio between the capture and projection frame rates. Increasing the camera frame rate corresponds to increasing the sampling frequency in time. By increasing the sampling frequency we obtain input sequences with more images than projected patterns. In the next experiment we want to evaluate which is the relation between the quality of the final 3D model relative to the frame rates ratio. Towards this goal, we have scanned a scene with varying camera frame rates and created a 3D model for each scan. Next, we have scanned the scene with the camera synchronized to the data projector and created a reference 3D model. We do not know how accurate the reference 3D model is in respect to the physical scene, thus, we do not call it ground-truth. Each 3D model was compared point-by-point to the reference model, the result is shown in Figure 12. We have also made a histogram of the distances for each model to better understand the distribution of the errors, also in Figure 12. We observed that models are more similar to the reference when increasing the camera frame rate, probably because of the increased sampling frequency, but even the model created with equal frame rates, Figure 12c, represents the scene with high detail. Figure 10 shows additional 3D models created with USL triangulated from the projector viewpoint and resolution. They have no visible difference with models created with SL, showing that SL and USL generates comparable results.

9.4 Global Illumination

We tried USL in the interior of a deep concave cup to see how interreflections on the internal walls would affect the scanning result. Figure 13 shows one of the images captured in the middle of a pattern change and a the binary image generated with USL. The figure also shows a color coded image representing the output of the SL algorithm with each color corresponding to one projector diagonal line. The last image in the figure is a render of the 3D model created by triangulation. We are interested on the middle and bottom regions at the right wall which receive direct light from the projector and reflected light from the bottom and sides of the cup. The region at the top was not illuminated by the projector. We see a number of decoding errors on these regions which could only be partially recovered. The decoding errors are caused by the misclassification of camera pixels as illuminated, even though they were on shadow, because of the light reflected from other regions. The problem is further discussed by Gupta et al. [Gupta et al. 2011] where they proposed to use high frequency patterns to reduce interreflections and low frequency patterns for semitranslucent surfaces. We scanned the same scene with synchronized SL with identical results.

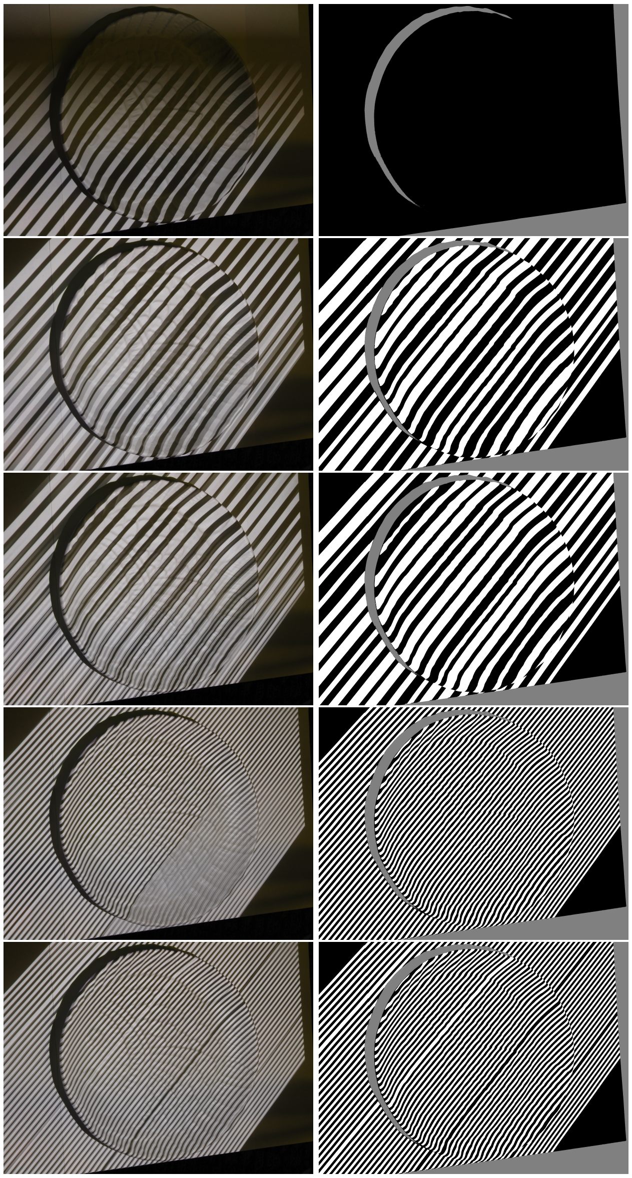


Figure 14: *(LEFT) Subset of a sequence of 15 images as captured by an unsynchronized rolling shutter camera running at equal framerate than a data projector. The example illustrates how captured images are a composition of two consecutive projected patterns, e.g. the top image has contributions from a totally black and a striped pattern. In addition, the amount of pattern contribution to each image varies from row-to-row because rolling shutter sensors do not capture all rows simultaneously. (RIGHT) Images estimated by our algorithm as they would have been captured by a synchronized global shutter camera. Zoom in to observe how all stripes are well defined and that each image corresponds to a single projected pattern.*

10 Limitations

We have presented a method for unsynchronized SL based on the image formation model of Section 6.2 which includes direct illumination terms and a global constant illumination C_p. It was seen during the experiments that it cannot predict well image intensities under the influence of varying global illumination effects, a limitation shared with its synchronized counterpart. Additional research is required to see if explicit separation of direct and global components [Nayar et al. 2006] and robust pixel classification [Xu and Aliaga 2007] can help to overcome this limitation as in SL, or if an extended formulation of USL including global illumination terms is possible. We are continuing our research in the last direction.

The time variables calibration presented here performed well in all experiments but increases the acquisition time because of the prepended subsequence. At this time, we rely on these extra patterns and we have manually selected the image subsequence and the image columns used for calibration. We are currently working on removing this requirement.

11 Discussion

In this paper we introduced the USL method, which enables off-the-shelf consumer digital cameras and smartphones to perform as high resolution high accuracy 3D snapshot cameras, and we have shown that it produces 3D data indistinguishable from the well known synchronized structured light method that it is designed to replace. USL only requires a SL flash to continuously project the sequence of patterns onto the scene while the camera captures images in burst mode. As future work we plan to design and fabricate a new structured light flash, in the form factor of a professional photographic flash or a smartphone docking station, which would turn on when triggered by the regular flash signal generated by the camera shutter, it would project the sequence of patterns, perhaps more than once to make sure that the images capture all the necessary data, and then it would stop. We also plan to explore USL to enable many camcorder, webcam, and video cameras used in consumer products and for industrial inspection, to function as 3D video cameras or 3D camcorders. In this case the SL flash continuously cycling through the patterns will be needed. The high bandwidth required between camera and processor, and the high computation power required to decode and stream the continuous sequence of 3D frames will be important implementation challenges.

12 Acknowledgments

The material presented in this paper describes work supported by the National Science foundation under the Grant No. IIP-1500249.

References

BELL, T., AND ZHANG, S. 2014. Towards superfast 3D optical metrology with digital micromirror device (DMD) platforms. In *SPIE MOEMS-MEMS*, ., 897907–897907.

BRADLEY, D., ATCHESON, B., IHRKE, I., AND HEIDRICH, W. 2009. Synchronization and rolling shutter compensation for consumer video camera arrays. In *CVPR Workshops*, IEEE, 1–8.

FUJIYOSHI, H., SHIMIZU, S., NISHI, T., NAGASAKA, Y., AND TAKAHASHI, T. 2003. Fast 3D position measurement with two unsynchronized cameras. In *CIRA*, vol. 3, IEEE, 1239–1244.

GODDYN, L., AND GVOZDJAK, P. 2003. Binary gray codes with long bit runs. *the electronic journal of combinatorics 10*, 1, R27.

GUPTA, M., AGRAWAL, A., VEERARAGHAVAN, A., AND NARASIMHAN, S. 2011. Structured light 3D scanning in the presence of global illumination. In *CVPR 2011*, 713–720.

HASLER, N., ROSENHAHN, B., THORMAHLEN, T., WAND, M., GALL, J., AND SEIDEL, H.-P. 2009. Markerless motion capture with unsynchronized moving cameras. In *CVPR*, ., 224–231.

HU, W., GU, H., AND PU, Q. 2013. Lightsync: Unsynchronized visual communication over screen-camera links. In *Proceedings of the 19th MobiCom*, ACM, 15–26.

INOKUCHI, S., SATO, K., AND MATSUDA, F. 1984. Range imaging system for 3D object recognition. In *Proceedings of the ICPR*, vol. 48, 806–808.

JAEGGLI, T., KONINCKX, T. P., AND VAN GOOL, L. 2003. Online 3D acquisition and model integration. In *Proc. IEEE Intl Workshop Projector-Camera Systems*, 4.

KOPPAL, S. J., YAMAZAKI, S., AND NARASIMHAN, S. G. 2012. Exploiting DLP illumination dithering for reconstruction and photography of high-speed scenes. *IJCV 96*, 1, 125–144.

LANGLOTZ, T., AND BIMBER, O. 2007. Unsynchronized 4D barcodes. In *Advances in Visual Computing*. Springer, 363–374.

MORENO, D., AND TAUBIN, G. 2012. Simple, accurate, and robust projector-camera calibration. In *3DIMPVT*, ., 464–471.

MORENO, D., SON, K., AND TAUBIN, G. 2015. Embedded phase shifting: Robust phase shifting with embedded signals. In *CVPR*.

NAYAR, S., KRISHNAN, G., GROSSBERG, M. D., AND RASKAR, R. 2006. Fast Separation of Direct and Global Components of a Scene using High Frequency Illumination. *ACM TOG* (Jul).

O'TOOLE, M., MATHER, J., AND KUTULAKOS, K. N. 2014. 3D shape and indirect appearance by structured light transport. In *CVPR*, IEEE Conference on, 3246–3253.

POSDAMER, J., AND ALTSCHULER, M. 1982. Surface measurement by space-encoded projected beam systems. *Computer Graphics and Image Processing 18*, 1, 1–17.

SALVI, J., PAGS, J., AND BATLLE, J. 2004. Pattern codification strategies in structured light systems. *Pattern Recognition 37*, 4, 827–849.

SALVI, J., FERNANDEZ, S., PRIBANIC, T., AND LLADO, X. 2010. A state of the art in structured light patterns for surface profilometry. *Pattern Recognition 43*, 8, 2666 – 2680.

SATO, K. 1987. Range imaging system utilizing nematic liquid crystal mask. In *Proc. 1st ICCV, 1987*, 657–661.

SHIRAI, Y., AND SUWA, M. 1971. Recognition of polyhedrons with a range finder. In *Proceedings of the 2nd IJCAI*, Morgan Kaufmann Publishers Inc., 80–87.

SRINIVASAN, V., LIU, H. C., AND HALIOUA, M. 1985. Automated phase-measuring profilometry: a phase mapping approach. *Appl. Opt. 24*, 2 (Jan), 185–188.

XU, Y., AND ALIAGA, D. G. 2007. Robust pixel classification for 3D modeling with structured light. In *Proceedings of Graphics Interface 2007*, GI '07, 233–240.

ZHANG, S. 2010. Recent progresses on real-time 3D shape measurement using digital fringe projection techniques. *Optics and lasers in engineering 48*, 2, 149–158.