

Fast Unsynchronized Unstructured Light

Chaima El Asmi, Sébastien Roy

Département d'Informatique et de Recherche Opérationnelle

Université de Montréal

Montréal (Québec), Canada

chaima.el.asmi@umontreal.ca, roys@iro.umontreal.ca

Abstract—This paper proposes a new approach in structured light correspondence to alleviate the camera-projector synchronization problem. Until now, great care was required to make sure that each camera image was corresponding exactly the correct pattern in the sequence. This was difficult to achieve with low-cost hardware or large size installations. In our method, the projector sends a constant video loop of a selected number of unstructured light patterns at a high frame rate (30 to 60 fps for common hardware), which are captured by a camera without any form of synchronization. The only constraint to satisfy is that the camera and projector frame rates are known. The matching process not only recovers the correct pattern sequence, but is impervious to partial exposures of consecutive patterns as well as rolling shutter effects.

Keywords- Unstructured light, Coded patterns, Projector-camera unsynchronized, 3D scanning.

I. INTRODUCTION

The first appearance of 3D scanners goes back several decades. For some time now, researchers have been looking for a way to scan objects and get a realistic 3D model. Nowadays, there are several 3D scanners based on different approaches to achieve 3D reconstruction. One is based on the projection of a laser beam and analyzes its trajectory as well as its deviation. Another one calculates how long a laser takes to get to the surface of the object and back to the projector, so-called *time-of-flight* [1]. Finally, some 3D scanner use structured light, which consists of projecting a known pattern encoding matching information onto an object and capturing so its geometry can be derived [2].

The most common method is to use a DLP projector and rely on Gray code patterns. They provide a temporal binary encoding of the position of each projector pixel. In this way, it is possible to establish the correspondence directly between the camera and the projector. However, a single error in a pattern will result in a wrong decoded position. A Gray code feature is that neighboring pixel codes differ only by a single bit. This spreads evenly the number of error bits across all possible codes [3]. However, the presence of low frequency spatial patterns in Gray code images increase the sensitivity to inter-reflections and indirect illumination [4], [5].

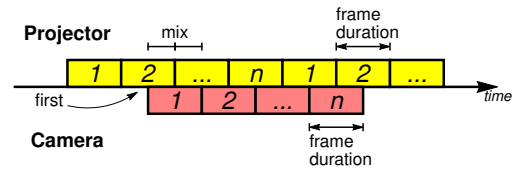


Figure 1. Synchronization model. The projector and camera share a common frame rate (fps). The first projector pattern seen by the camera can be any pattern. Exposure time and temporal overlap result in a mixture of two consecutive patterns in a single camera image.

A. Unstructured Light

Unstructured light is also a projected encoding, but it does not directly encode the pixel positions [6]–[9]. The temporal sequence provides a binary code which is then matched to a reference sequence. The decoupling of encoding and patterns makes it possible to select the spatial frequency and achieve great robustness to indirect lighting as well as difficult capture conditions. In this paper, the complete flexibility in the selection of unstructured patterns will prove essential to solve the synchronization problem. An important advantage is that it is possible to make an unstructured light correspondence from the projector to the camera as well as the usual camera to the projector matching. This alleviates the need to compute an inverse mapping, which is prone to errors.

B. Synchronization

The synchronization of the camera-projector system can take multiple forms. In this section, we will describe the three most important aspects for our method: first image, temporal offset, and frame rate, as illustrated in Fig. 1. First, the projector sends a video loop repeating n patterns and the camera starts capturing at an arbitrary time. The first image in the sequence is therefore an unknown pattern and must be estimated (Fig. 1, first). To make this problem simple, some methods project special patterns, or repeat the first pattern, in order to allow easy identification of the start of the sequence [10]. This will not be necessary in our case. Secondly, there is a temporal offset between the camera and the projector image sequences (see Fig. 1, mix). This offset generally results in a mixture of two consecutive patterns in a single captured image. The multiple exposure

problem makes the decoding or establishing correspondence between camera and projector images impossible. Aside from using synchronization, the multiple exposure problem can be solved with an accurately calibrated image formation model [11] or by designing the method to be robust to this effect, as we do in this paper. Finally, the last aspect is the frame rate of the camera and the projector (see Fig. 1, frame duration). Both frame rates are assumed to be known in advance, and to never change in time. It is possible to account for differences in frame rates, but the simplest and most practical approach with common hardware is to enforce the same frame rate for both the camera and projector, thereby ensuring stable multiple exposure of consecutive frames. When frame rates are not matched, we can end up with many duplicate images of a single pattern, or missing patterns. These two cases, even if they can be alleviated with some effort, are easy to avoid in practice, since frame rates tend to be standardized and stable.

In this article, we will explain our method, proceed to identify the first pattern, and then establish correspondence with the reference patterns while discovering the mix of multiple exposures.

II. PREVIOUS WORK

There has not been much work on the specific problem of unsynchronized structured light. Most methods assume perfect synchronization of the camera-projector system or try to improve it. The article [12] summarizes well the three main approaches of this problem: hardware synchronization, software synchronization, and no synchronization. The first one is hardware synchronization. It is a triggering circuit that synchronizes the projector and the camera [13]–[15]. This approach is not accessible to everyone because it requires hardware, namely a projector and a camera that each supports external synchronization. This is rare for inexpensive hardware. Its advantage is that the system can scan at a very high frame rate (sometimes up to 3000 fps) [16], [17]. The second approach is to synchronize the camera-projector system by software [10], [18], [19]. Unlike the previous one, it does not cost as much and allows common off-the-shelf hardware, but it requires a longer acquisition time [20]. It must ensure that each projected pattern is captured by the camera without a risk of multiple exposure of consecutive patterns. The resulting scan time is not acceptable in many cases, such as scanning faces. In such situations, the acquisition time has to be reduced because a person cannot stay still more than a few seconds. Also, since we intend to use low-cost hardware, no synchronized solution is adequate. That's why we propose the unsynchronized approach.

There are some methods that are based on the unsynchronized approach [11], [21], [22]. One successful method that is close to ours is [11], where a detailed image formation model is devised and solved to recover synchronization parameters and subsequently recover Gray code matching

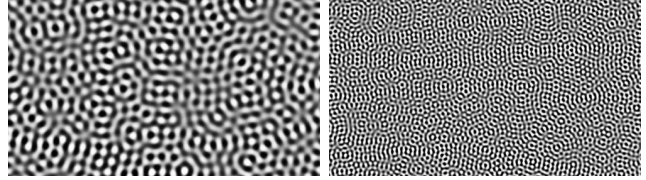


Figure 2. Unstructured light patterns can be chosen arbitrarily. Above, two such patterns at different spatial frequencies.

patterns. Their algorithm relies on exact knowledge of the image formation process to be able to determine the time required by the camera to capture each single row of the projected image (they assume a rolling shutter camera), as well as the time required to capture the entire image. In this way, they can resolve the multiple exposure problem present in the unsynchronized captured images. To detect the first projected pattern, extra frames are added at the beginning of the sequence. They are either full black (B) or full white (W) frames, set up as the sequence {B, B, W, W, B} so it becomes easy to detect it in the camera image sequence.

Overall, their method is quite complex computationally, especially given the equation systems that have to be resolved, and its reliance on Gray code patterns makes it more sensitive to inter-reflections. However, their results indeed prove that unsynchronized capture can be done, even if the hardware requirement is not completely general and require a special fast projector.

In this paper, we present a very simple and fast method. Described in Sec. III, it scans without synchronization at 30 to 60 fps in less than two seconds. Moreover, it does not require any special hardware, as the experiments presented in Sec. IV will illustrate.

A. Generating unstructured light patterns

In this paper, we rely on the concept of unstructured light patterns presented by Couture *et al* [6]. However, the patterns are generated in a different way, such that they don't present large white or black regions. They are built simply as a sum of sines with random orientations but similar frequencies, so each unstructured light pattern is different. Fig. 2 shows two such patterns at different frequencies. The quality of the unstructured match depends on the spatial frequency and the number of bits of code, which itself depends on the number of patterns. If the frequency is too low, then the codes of neighbouring pixels become too similar which causes bad matches. Increasing the number of patterns can help remove this ambiguity. Using higher frequency patterns can also help (see Fig. 2, right), but if the frequency becomes too high then the camera might not be able to distinguish light and dark bands, resulting in matching errors.

B. Quadratic code

Our method will use linear code to find the first image of the sequence, and then quadratic code to estimate the mixture of the captured images and to compute the final matching.

As demonstrated by Couture *et al* [23], for n patterns, the linear code-length is n bits long and provides n bits of information. On the other hand, the quadratic code is derived from the same n patterns, have a length of $\frac{n^2-n}{2}$ bits, which provide $n \log n$ bits of information, the maximum possible out of n patterns. As an example, 30 patterns can provide a linear code of length 30 bits for 30 bits of information, or a quadratic code of length 435 bits for 147 bits of information.

Establishing pixels correspondence between camera and projector is accomplished by the LSH (Locality Sensitive Hashing) algorithm [24]. This probabilistic algorithm is specialized in searching for nearest neighbors in very high-dimensional spaces. At each iteration, it generates different match proposals from which we keep only the best one at the end. Because of its probabilistic nature and efficiency in high dimensions, LSH is better suited to match long codes. That's why, in order to increase the amount of information and decrease the matching error, we prefer using quadratic codes with fewer images, so the capture time remains small.

III. UNSYNCHRONIZED CAMERA-PROJECTOR SYSTEM

As illustrated in Fig. 1, a projector plays a continuous video loop of a number of patterns (typically 30 or 60) at a fast frame rate, typically the same as the projector patterns. The patterns are random unstructured light patterns. The capture starts somewhere in the sequence without synchronization. The camera and the projector are assumed to have the same frame rate, so each projected pattern is seen exactly once, possibly mixed with another one, by the camera. In practice, it is important to give enough time to the camera to adjust itself, so we skip a number of images at the start of the capture. Also, when scanning deformable or moving objects such as faces, capturing more images can be helpful since it allows us to choose the optimal sequence where the person moves the least. As we know, it is very hard for a human to remain still for more than two seconds. This is why we focus on scans that require two seconds or less.

Since the start of the camera capture is unsynchronized, it is essential to establish which pattern is seen first. Since we capture the same number of images as there are projected patterns, all the algorithm has to do is search across all captured images for the first pattern. However, it is impossible to recognize such a pattern, so another approach must be used.

A. Finding the first image of the sequence

As mentioned previously, the camera can start capturing at any time while observing a continuous video loop of patterns. The first step of our method is thus to find which

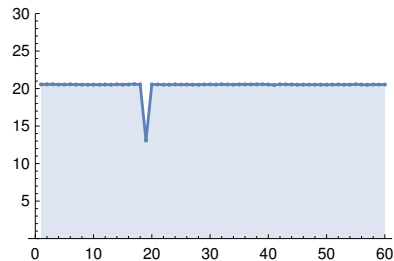


Figure 3. Matching costs for finding the first image. The average of the erroneous bits decreases from 22 to 13 on a 60 bits code. The x and y axes represent the pattern number and the number of erroneous bits, respectively.

image in the captured sequence corresponds to the first pattern. For this section, we assume that the images are not a blend of partially exposed consecutive patterns. Even if this is not true in practice, partial multiple exposures always feature a "dominating" pattern, which is enough for finding the first image.

First, we compute a binary code for the projector pattern sequence and the captured image sequence in their original order, both from their start. For speed purposes, the code is linear, not quadratic. Then, we compute a first correspondence between the projector and the camera, by running a small number of iterations of LSH (typically 6). The intent is not to obtain an actual match but just to detect if we are matching completely unrelated images or not. We calculate the sum matching costs after a few iterations of LSH. After that, we shift the camera codes to the left, effectively changing which image is considered the first, while the projector codes remain unchanged. We then proceed to match these codes again and keep the matching costs. After n shifts, we have tested all possible matches. All matches should have a similar high cost, and a single match will be lower, indicating we have found the first image. Once the minimum is found, we reorder the initial camera sequence according to its position found and the sequences are effectively aligned.

For this step, we use linear code because the gap between the sum of the first image and the other sums is large, even after the first iterations of LSH, as shown in Fig. 3. The matches obtained are affected by the partial exposure of the camera, but the gap between the sums is not really affected. That's why we have assumed above that our images are perfectly exposed. In the next section, we will introduce partial exposure that affects our captured images.

B. Partial exposure of consecutive patterns

When capturing asynchronously, as illustrated in Fig. 1, it is entirely possible that the projected pattern will change during the exposure of a single image. The resulting image will feature a multiple exposure, a mixture, of two consecutive projected patterns. When this happens, matching will become harder, or even impossible. This problem is even

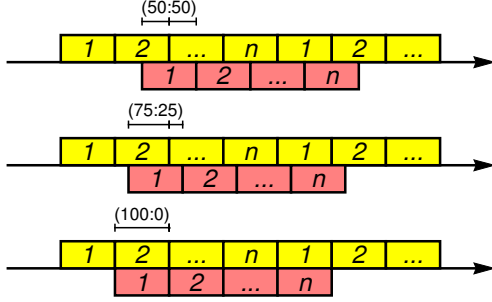


Figure 4. Different mix values (in percentage) that represent three cases of a partial exposure (50%, 75% and 100%) of two consecutive patterns. The last image represents a rare case of a synchronized capture without mixture (bottom).

worse for rolling shutter cameras, since the exposure start time varies from the top to the bottom of the camera image. In these conditions, matching with the original projected patterns will yield a partial match, at best, as illustrated in the top left of Fig. 5.

In the presence of partial exposures, the *finding first image* step previously presented will still work. It will pick the *strongest* exposure as the reference for a given image. However, even with the correct first image, the actual matching must account for partial exposure.

Unstructured codes, contrary to structured codes, allow total freedom in the selection of the projected patterns. This property will now become very useful. Instead of considering an image as a partial exposure of two consecutive patterns, we will consider that the image is a full exposure of a new reference pattern which is built from a mixture of two consecutive patterns. Fig. 5 illustrates the matching results obtained for various mixing of consecutive patterns.

The method for estimating this "consecutive pattern mixture" is described in the following section.

C. Mixture of captured images

As defined above, the lack of synchronization can induce a multiple exposure between two consecutive patterns, as illustrated in Fig. 4. The first and second cases (top and middle) are a typical variation of the exposure between the camera and the projector. This variation generates the mixture of two consecutive patterns. The mixture represents a percentage (50:50 or 75:25) of each projected pattern. The last example is a rare case (bottom) where the frame rates of the camera and the projector are synchronized, so the camera capture a perfect exposure of only one pattern.

If the frame rate durations of the camera and the projector are not matched, then the mixture will change in time. When such a mismatch is present, various temporal mixing strategies can be used, but at an increased computation cost. Considering that frame rates are usually stable in time, it is easy to devise a way to measure those frame rates

beforehand. Then, it is much easier to take these different frame rates into account in the matching method. In practice, when the camera and projector frame rates are similar enough (i.e. 60 fps and 59.94 fps), the variation in mixture over a few seconds is negligible. This is our assumption in this paper's experiments.

For a global shutter (i.e. progressive) camera, we expect that a single mix value will be used across the whole camera image. We could then *search* for the best mix value and then proceed to match with that value. On the other hand, for a rolling shutter camera, the mix changes vertically across the camera image.

In our method, we will allow one mix value per pixel, so rolling shutter as well as global shutter camera are supported. However, we assume that the mix value does not change in time during the capture. This implies that the frame rates of the camera and projector are well matched.

First, we must find the two consecutive patterns that are mixed in the captured images. The I_i image can be mixed with the previous pattern ($i - 1$) or the next one ($i + 1$), so we must evaluate both cases. We calculate the error of the two cases over a few iterations of LSH. The smallest error indicates which of the two cases (lets call them $d = -1$ and $d = +1$) is the one contributing to the mixture of images.

From now on, we will explore a number of possible mixtures of patterns i and $i + d$. For each tested mixture μ , we will generate a new set of reference images $I'(i) = \mu I(i) + (1 - \mu)I(i + d)$. The camera image sequence will be matched to these reference patterns and any resulting good match will be kept. In practice, we test the set of mixtures from 0 to 1, with a step of 0.1. For each pixel, its final match will be the one with minimal cost over all these mixtures. This not only provides robustness to the distribution of mixtures in the image, but provides a simple estimation of that mixture, as illustrated in Fig. 9.

For this part, matching relies on quadratic codes, because we want the maximum amount of information for each pixel to increase the quality of the match. For the same reason, we also increase the number of iterations of LSH (typically 80 iterations). In the bottom of Fig. 6, the matching result is provided as a Lookup Table that we obtained after finding the first image and the mix. As an illustration of the effectiveness of this approach, we illustrated in Fig. 5 separate results for different mixtures and a rolling shutter camera. It can be clearly seen that each mixture yields an associated correct match, and combining these matches will result in a correct solution. Contrary to [11], the optimal mixture is obtained without explicitly solving an internal camera imaging model.

IV. EXPERIMENTS

In this section, we present the experimental evaluation of our proposed method on real scenes. Since all the current

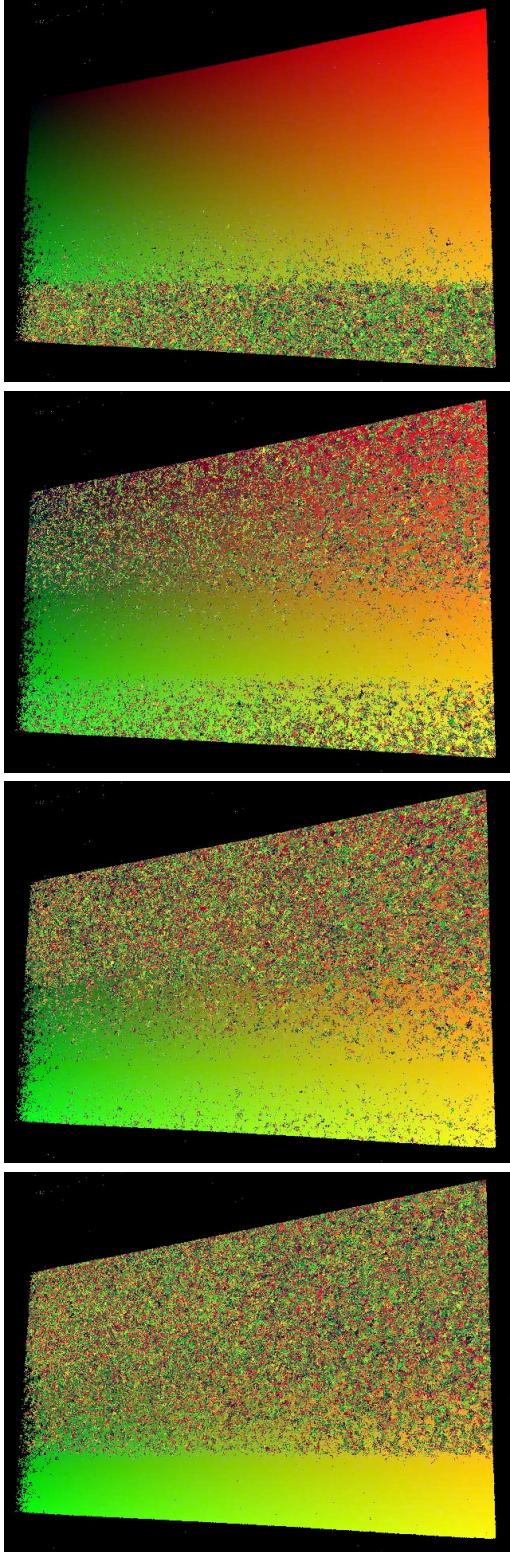


Figure 5. Matching obtained for various partial exposures of two consecutive patterns. In each matching, x and y coordinates are represented as red and green, respectively. The mixtures illustrated are 0, 0.5, 0.6, and 1.0

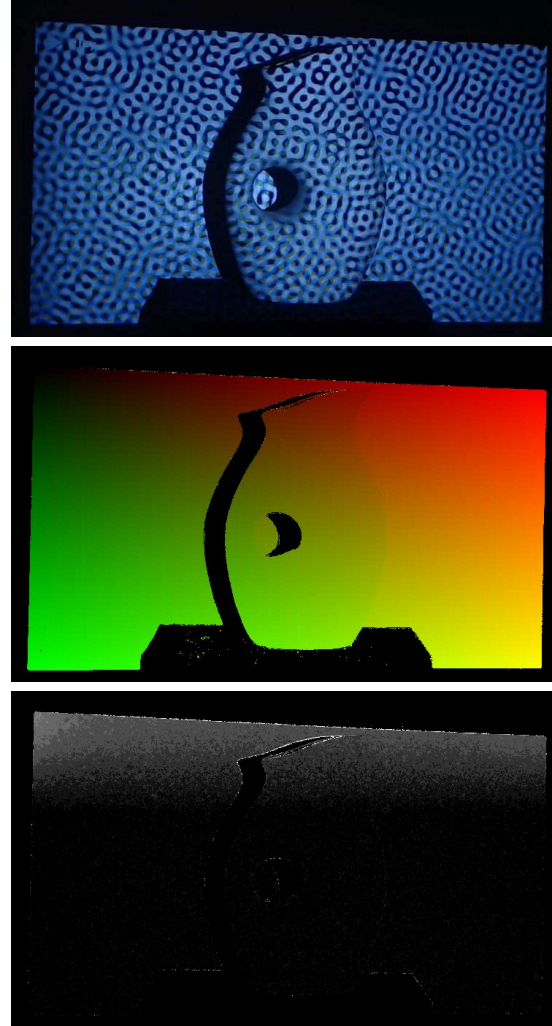


Figure 6. Projected patterns on a typical scene, observed by the camera (top), camera-projector Lookup Table (middle) where red and green represent x , y positions, and computed mix of successive patterns (bottom). The mix value changes from top to bottom, indicating a rolling shutter camera.

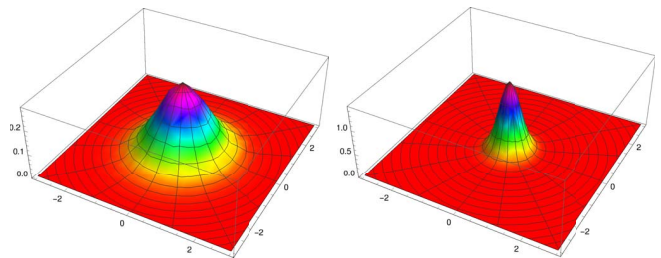


Figure 7. Average displacements (in pixels) between unsynchronized matching and synchronized matching (left), and for running twice synchronized matching on the same images (right). The maximum precision on the right is limited by the number of LSH iterations.

| fps | scene | N | std | std ref | loss |
|-----|-------|-----|------|---------|------|
| 30 | plane | 30 | 0.65 | 0.22 | 0.43 |
| | | 60 | 0.49 | 0.29 | 0.20 |
| | | 120 | 0.49 | 0.04 | 0.45 |
| | vase | 30 | 0.86 | 0.23 | 0.63 |
| | | 60 | 0.66 | 0.28 | 0.38 |
| | | 120 | 0.35 | 0.04 | 0.31 |
| 60 | plane | 30 | 0.76 | 0.34 | 0.42 |
| | | 60 | 0.55 | 0.17 | 0.38 |
| | | 120 | 0.41 | 0.09 | 0.32 |
| | vase | 30 | 0.77 | 0.34 | 0.43 |
| | | 60 | 0.55 | 0.15 | 0.40 |
| | | 120 | 0.81 | 0.08 | 0.73 |

Table I

THE STANDARD DEVIATION OF DISPLACEMENTS (IN PIXELS) IN x, y OBTAINED WITH A DIFFERENT NUMBER OF PATTERNS AND A DIFFERENT FPS ONTO TWO SCENES. STD REPRESENTS THE STANDARD DEVIATION OF DIFFERENCES BETWEEN SYNCHRONIZED AND UNSYNCHRONIZED. STD REF REPRESENTS THE STANDARD DEVIATION BETWEEN TWO SYNCHRONIZED SCANS. LOSS IS THE DIFFERENCE BETWEEN THE TWO STANDARD DEVIATIONS.

methods that need to synchronize the projector-camera system use structured light, we can't compare them directly with our method because we use unstructured light patterns. So we made synchronized and unsynchronized matches to assess the accuracy and quality of the unsynchronized method itself, not the LSH algorithm. We provide results for different number of patterns, different frame rates, for two different scenes. The first set of results use quadratic code and the second set use linear code. The goal of our method is to make possible a 3D scanner with the simplest camera-projector hardware possible. For the realization of all our experiments, we used non-professional hardware. The projector was an Aaxa HD Pico projector used at its native resolution of 1280x720 for all scans, both at 30 and 60 fps. Two cameras were used, at different frame rates. A Logitech C920 webcam was used at 1920x1080 resolution at 30 fps (its maximum rate) and a GoPro HERO3+ was used at a resolution of 1920x1080 for scans at 60 fps. Both cameras are rolling shutter with automatic brightness adjustment, the so-called *auto gain*. This type of camera can cause a problem during a scan because it tries to adapt to lighting changes throughout the capture. Fortunately, unstructured light patterns are notorious for their constant average intensity, so they are well suited for this kind of camera. Finally, we encountered a problem of flickering during the experiments. This flicker depends on the light source observed. Most ordinary projectors will create RGB color images by presenting the RGB planes one color after another. When the exposure time is short, a camera sees these individual color planes as a flicker (the *color wheel* effect). It is thus important to ensure that the camera

exposure time is as long as the frame rate will allow ($\frac{1}{60}$ sec is perfect for most projector).

In our experiments, we projected continuous video loops of 30, 60 or 120 patterns. For a first set of experiments, they were projected at 30 fps on two scenes ("plane" and "vase") and observed at 30 fps by the Logitech camera. For a second set of experiments, they were projected at 60 fps and observed at 60 fps by the GoPro camera. A final set of "synchronized" experiments was also performed with projection at 5 fps, observed from both cameras, to obtain reference matches for comparison.

The results are provided in Table I.

In order to evaluate only the performance of the synchronization aspect of our method, we compared unsynchronized matchings with synchronized matchings. This ensures that we do not compare the structured light method itself, but only the impact of synchronization. The column "std" indicates the standard deviation in pixels observed between unsynchronized and synchronized results, in pixels, in the camera reference frame at 1920x1080 resolution. We observe deviations of fewer than one pixel in all cases, which illustrates how well the method works.

However, since the matching algorithm is probabilistic (LSH), its solution varies. The column "std ref" measures this variation by comparing two synchronized matching obtained on the same captured images. These values, which are generally very small, constitute the precision limit of the method (at the selected number of LSH iterations). This implies that the true *loss of precision* resulting from removing synchronization is in column "loss", expressed as the difference between the deviations of the two columns "std" and "std ref".

In analyzing those results, one must consider that the matchings are all computed with integer pixel positions. This artificially increases the standard deviations when it is lower than one pixel. A new set of experiments should be devised eventually to use sub-pixel matching, since the observed accuracy seems to justify it.

Overall, the results are very good and illustrate that the loss of precision for our method is minimal.

Typical distributions of matching errors, measured as the number of different bits between codes, are illustrated in Fig. 7. The curve on the left corresponds to the differences of a synchronized scan and an unsynchronized one. The curve on the right corresponds to comparing two synchronized scans run on the same captured images. We observe that the distributions are approximately normal with a larger standard deviation for unsynchronized matching, as expected.

For the latest experiments, we used the same sets of patterns and captured images (60 and 120 patterns projected onto a "plane") to generate the Lookup Table using a linear code. Fig. 8 illustrates the errors distribution for a synchronized and an unsynchronized matching. They are very similar, indicating that the lack of synchronization does

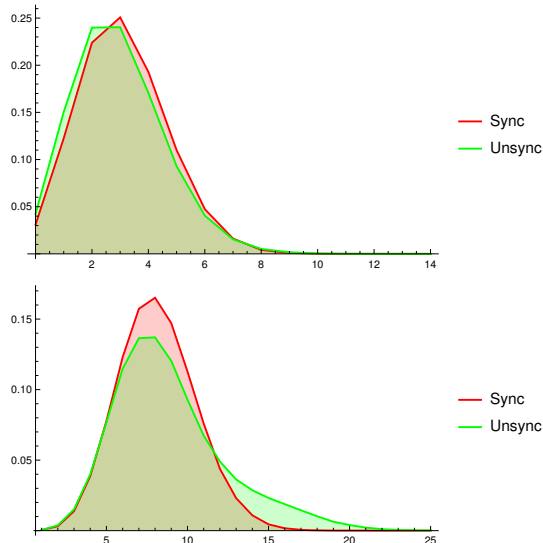


Figure 8. Curves representing the probability density of matching errors, measured in bits, for unsynchronized matching (green) and synchronized matching (red). These experiments were generated with 60 (top) and 120 (bottom) patterns and matched with a linear code.

not make the matching more difficult.

Notice that the synchronized distributions are binomial, with n trials corresponding to the linear code-length in bits (here 60 and 120) and a probability p related to the capture conditions. The unsynchronized distributions (especially with 120 bits) are modeled as a mixture of two binomials, one being the same as the synchronized case, and one with a slightly larger p which applies for more difficult mixed exposure cases. Overall, the variation between the two curves in Fig. 8, synchronized in red and unsynchronized in green, are small.

One of the goals of our method is to make scanning faces easier. Scanning in less than two seconds makes this much easier. Fig. 9 shows the different mix values obtained while scanning a face, illustrating that the method works well for these kind of applications. Adding calibration and sub-pixel accuracy to the matching process, we expect to obtain excellent 3d models of faces.

V. CONCLUSION

In this article, we presented a novel unsynchronized active scan method based on an unstructured light framework. Tested at the limit of current low cost cameras and projectors, at 30 and 60 fps, we obtained results that are extremely close to what can be obtained with synchronization. We expect this method to make scanning in some difficult situations much more feasible, such as scanning human faces or large objects where the camera is too far from the projector to be easily synchronized. In the future, it should be straightforward to add sub-pixel accuracy to the matching algorithm, make it faster, parallel and fully test



Figure 9. Selected best mix values for a rolling shutter camera. The mix values range from 0 to 1, corresponding to increasing levels of partial exposures between consecutive patterns. The image on the top represents a scan seen by the camera and the image on the bottom by the projector.

its performance on full 3D reconstructions. Also, using a custom fast projector and industrial camera, it should be possible to achieve very high capture rates, which could open new applications.

REFERENCES

- [1] F. Blais, J.-A. Beraldin, S. El-Hakim, and G. Godin, "New development in 3d laser scanners: From static to dynamic multi-modal systems," in *6th Conf Opt 3-D Meas Tech*, 2003, pp. 22–26.
- [2] J. Salvi, J. Pagès, and J. Batlle, "Pattern codification strategies in structured light systems," *PATTERN RECOGNITION*, vol. 37, pp. 827–849, 2004.
- [3] S. Inokuchi, "Range imaging system for 3-d object recognition," *ICPR, 1984*, pp. 806–808, 1984.
- [4] S. K. Nayar, G. Krishnan, M. D. Grossberg, and R. Raskar, "Fast separation of direct and global components of a scene using high frequency illumination," *ACM Transactions on Graphics (TOG)*, vol. 25, no. 3, pp. 935–944, 2006.
- [5] J. Gühring, "Dense 3d surface acquisition by structured light using off-the-shelf components," in *Videometrics and Optical Methods for 3D Shape Measurement*, vol. 4309. International Society for Optics and Photonics, 2000, pp. 220–232.
- [6] V. Couture, N. Martin, and S. Roy, "Unstructured light scanning to overcome interreflections," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1895–1902.

- [7] A. Kushnir and N. Kiryati, "Shape from unstructured light," in *3DTV Conference, 2007*. IEEE, 2007, pp. 1–4.
- [8] Y. Wexler, A. W. Fitzgibbon, and A. Zisserman, "Learning epipolar geometry from image sequences," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 2. IEEE, 2003, pp. II–209.
- [9] J. Davis, R. Ramamoorthi, and S. Rusinkiewicz, "Spacetime stereo: A unifying framework for depth from triangulation," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 2. IEEE, 2003, pp. II–359.
- [10] T. Jaeggli, T. P. Koninckx, and L. Van Gool, "Online 3d acquisition and model integration," in *PROCAMS, ICCV Workshop, 2003*.
- [11] D. Moreno, F. Calakli, and G. Taubin, "Unsynchronized structured light," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 6, p. 178, 2015.
- [12] T. Petković, T. Pribanić, M. Đonlić, and N. D'APUZZO, "Software synchronization of projector and camera for structured light 3d body scanning," in *7th International Conference on 3D Body Scanning Technologies, 2016*.
- [13] I. Ishii, K. Yamamoto, K. Doi, and T. Tsuji, "High-speed 3d image acquisition using coded structured light projection," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*. IEEE, 2007, pp. 925–930.
- [14] K. Liu, Y. Wang, D. L. Lau, Q. Hao, and L. G. Hassebrook, "Dual-frequency pattern scheme for high-speed 3-d shape measurement," *Optics express*, vol. 18, no. 5, pp. 5229–5244, 2010.
- [15] Y. Wang, K. Liu, Q. Hao, D. L. Lau, and L. G. Hassebrook, "Period coded phase shifting strategy for real-time 3-d structured light illumination," *IEEE Transactions on Image Processing*, vol. 20, no. 11, pp. 3001–3013, 2011.
- [16] J. Takei, S. Kagami, and K. Hashimoto, "3,000-fps 3-d shape measurement using a high-speed camera-projector system," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*. IEEE, 2007, pp. 3211–3216.
- [17] S. Zhang, D. Van Der Weide, and J. Oliver, "Superfast phase-shifting method for 3-d shape measurement," *Optics express*, vol. 18, no. 9, pp. 9684–9689, 2010.
- [18] O. V. Olesen, R. R. Paulsen, L. Hojgaard, B. Roed, and R. Larsen, "Motion tracking for medical imaging: a nonvisible structured light tracking approach," *IEEE transactions on medical imaging*, vol. 31, no. 1, pp. 79–87, 2012.
- [19] T. P. Koninckx and L. Van Gool, "Real-time range acquisition by adaptive structured light," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 3, pp. 432–445, 2006.
- [20] K. Herakleous and C. Poullis, "3dunderworld-sls: An open-source structured-light scanning system for rapid geometry acquisition," *arXiv preprint arXiv:1406.6595*, 2014.
- [21] R. Sagawa, R. Furukawa, and H. Kawasaki, "Dense 3d reconstruction from high frame-rate video using a static grid pattern," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 9, pp. 1733–1747, 2014.
- [22] Z. Zhang, "Microsoft kinect sensor and its effect," *IEEE multimedia*, vol. 19, no. 2, pp. 4–10, 2012.
- [23] N. Martin, V. Couture, and S. Roy, "Subpixel scanning invariant to indirect lighting using quadratic code length," in *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1441–1448.
- [24] A. Andoni and P. Indyk, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," in *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*. IEEE, 2006, pp. 459–468.