Fully in tensor computation manner: one-shot dense 3D structured light and beyond

Xuan-Li Chen\textsuperscript{1,}*, Luc Van Gool\textsuperscript{1,2}
\textsuperscript{1}PSI-ESAT, KU Leuven, Belgium
\textsuperscript{2}CVLab, ETH Zurich, Switzerland
*E-mail: xuanli.chen@icloud.com

Abstract: Tensor computation evolves fast towards a prosperous existence in recent years, e.g. PyTorch. An immediate advantage of using tensor computation is that one does not need to implement low-level parallelism to attain efficient computation, which is of simplicity for both research and application development. The authors began with discovering that a simple manoeuvre ‘tensor shift’ could perform neighbourhood manipulation in very efficient parallel manner. Based on ‘tensor shift’, they derive the tensor version of a renowned correspondence search algorithm: semi-global matching (SGM), which they prefix the name as tensor-SGM. To evaluate their idea, they build-up a novel and practical one-shot structured light 3D acquisition system, which yields state-of-art reconstruction results using off-the-shelf hardware. This is the first fully tensorised 3D reconstruction system published to the authors’ best knowledge, and it opens new possibilities. A major one is, in the same tensorised framework, they solved the pattern interfering problem which hinders multi-structured light systems from working together. This part is marked as ‘beyond’ in this study to avoid confusing the readers the spotlight: the fully tensorised 3D structured light framework.

1 Introduction

Computation efficiency is now-a-days fundamental requirement, and several parallel computing development kits are released (e.g. different GPUs/TPUs/’AI Chips’ with their tool-kits).

As tools, not only the running performance but also the corresponding learning curves matter. Basically, one has to spend weeks to learn one parallel pipeline and months before venturing to productive development for either research or industry (e.g. smart phone apps). Few people could afford to devote such long time to build-up this skill, and even the skill itself is rather spontaneous as we are in a time in which computation platforms seem to evolve.

Industry is making tremendous endeavour to provide better computing products. Tensor computation platforms (e.g. PyTorch\textsuperscript{1}, Tensorflow) come into existence and provide a high level API to access this effective computation power, which renders possible an efficient access to tame parallel computing. A complete course (e.g. tutorial from corresponding site) allows one to get started in a rather short time (a few days, depends on one’s previous programming experience). In other words, it increases user friendliness for rigorous algorithms. On the other hand, extensible capability is attained as these platforms seem to bridge also hardware gaps during development (e.g. clouds, clusters, desktop GPUs and mobile devices). In short, companies (e.g. Nvidia, Cambricon \textsuperscript{2}etc.) release evolving hardware to support tensor computation platforms to keep improving the computational performance.

Computer vision, as an engineering science, is influenced not only by the correctness of the theory but also the computing to approximate the solution. The first part of our paper is based on semi-global matching (SGM)\textsuperscript{[3]}, which is an good example. SGM makes significant contributions to both academy and industry by making dense corresponding search computation-wise plausible. Its main innovation is a strategy to approximate the global cost by combining sub-cost of multiple paths, whose computation is effectively carried out by dynamic programming. It makes sense to migrate successful computer vision algorithms to tensor platforms where developers can deploy parallel computation power with ease, and at the meanwhile, the simplified work-flow of codes (as less low-level parallel commands involved) can help the community to robustly re-produce the reported results (less local hardware configuration dependent) from other peers. Besides, tensor platforms carry now-a-days popular deep learning models for recognition purposes. Implementing the full stack system on a single platform is beneficial.

In this work, we derive and implement SGM, in tensor form, based on which we build a one-shot dense 3D acquisition system using structured light, which is novel, fast and capable to produce state-of-art reconstruction results. As a new approach, it opens new possibilities as we expected. We went beyond soon. The same tensorised framework help construct a solution to patterns interfering problem, which prevents multiple one-shot structured light systems to working together. We mark the work coping multiple structured light systems with ‘beyond’ throughout the paper, so that to remind the readers that the main progress we would like to present here is: the fully tensorised 3D structured light framework.

1.1 Brief the structured light system

The main part of this paper describes our work building a ‘one-shot’ 3D acquisition system using structured light and fully implemented in tensor form. Fig. 1 shows the appearance and Fig. 2 the work-flow.

Being an active illumination approach, our system is suitable for texture-less surfaces, where multi-view stereo approaches struggle. ‘One-shot’ property shields our system against motion that may otherwise be substantial during capturing, thus also enabling the capture of dynamic 3D, in quick succession (4D reconstruction) (cf. Section 8.3). Our system is fully designed and implemented in tensor manner, so that to deploy parallel computation power with ease.

An important finding during our work is that the ‘tensor shift’ manoeuvre (cf. Section 6.1) access neighbours for the full-scale image in parallel, which can apply to other algorithms with similar characteristics, e.g. random fields.
1.2 Attenuation measurement

To establish dense (per-pixel) disparity map therefore 3D reconstruction, we deploy SGM. Yet, the matching cannot perform well directly between projected pattern and captured image. Though theoretically one can treat a projector as an inverse camera, the projected pattern itself does not contain any photometric information. Or to say, a pattern from projector to being captured by camera is an information loss process. Empirically, this process behaves like a low-pass filtering, i.e. low frequency part seems to endure while the high frequency part lost, a blurred image. We find that by measuring the spectrum attenuation and generate patterns using according bandpass spectrum, the SGM can establish correspondence with high accuracy and robustness. Therefore, in our experiment we measure the attenuation ratio across spectrum bands before generating texture patterns. This is beneficial, as details beyond certain threshold would simply disturb the correspondence search instead of contributing (projector ‘sees’ these details while camera not). To make the measurement convenient, we proposed a method with invariance given epipolar geometry [4] and discrete cosine transformation (DCT) horizontal constant spectrum [5]. Details are given in Section 3.

1.3 Our projected patterns

The pattern is crucial in structured light, and in general active illumination based 3D reconstruction methods. Instead of inserting handcrafted ‘codes’ to disambiguate sparse coordinates via their spatial surroundings [6], our pipeline retrieves 3D information by establishing dense correspondences between the captured image and projected pattern. As usual, we exploit epipolar geometry to simplify the process. The actual 3D coordinates then follow from triangulation. The pattern that we project consists of high-contrast noise, thus creating locally unique texture patterns (i.e. the ‘implicit code’ of the pattern is the texture itself [7]). The noise is of sufficient frequency to localise the patterns, necessary to survive the change in viewpoint between projector and camera. The spectrum of our pattern is generated randomly in a bandpass range, as a band-pass random noise pattern. It is governed by one bandwidth parameter measured previously, that steers the roughness of the pattern, so that patterns at different resolution and frequency can be generated as long as they are in proper range. More details are provided in Section 4.

1.4 Correspondence search

For the correspondence search we propose tensor-SGM (tSGM) [8], innovated from SGM [3], which searches for the disparity map that minimises an approximated global cost between two images with known epipolar geometry (cf. Section 5.2). For the inter-pixel metric, we chose the census transform [9], which calculates a binary vector for each pixel encoding their topological relation regarding local neighbourhoods. Details are given in Section 5.

1.5 Derive SGM to tSGM

A cornerstone of our proposed system is ‘tensor shift’ manoeuvre, which enables the tensorisation of SGM into tSGM [8], so that to be easily cast into parallel computing devices using tensor processing libraries (PyTorch in our work). Concrete derivation
and examples are in Section 6. To demonstrate the overall performance of our system, several 3D scene reconstructions in real-world are performed (cf. 8), and a 4D reconstruction (dynamic capturing) is given in the video corresponding to this paper. We further demonstrate that the ISGM framework is not only suitable for SGM itself, but also can be easily extended into its variants; details are given in Section 8.1.1.

1.6 Experiments
We perform experiments in three perspectives: (i) “tensor shift” can lead to more innovations (cf. Section 8.1); (ii) quality of 3D reconstruction (cf. Section 8.2); and (iii) 4D reconstruction (cf. Section 8.3).

1.7 Beyond: cooping multiple structured light systems
Same in tensor manner, we develop a framework to enable multiple structured light units to work simultaneously, which support applications require detail 3D from different profiles, e.g. in-situ face identification. The key is to let the camera know where the patterns come from when more than one patterns present. Then follows the reconstruction strategy: a projected pattern in combination with a camera allows for a structured light approach, which reconstructs regions illuminated by a single pattern; overlapped patterns often indicates being captured by both cameras, which allows active stereo. To identify single pattern regions versus overlaps, we watermark the different projection patterns while preserving enough textures for correspondence match. In the setup reported by this paper, two projectors and two cameras are deployed. The same mechanism can be extended to many.

1.7.1 Spectrum embedding: We develop our method in spectrum embedding/watermarking, whose characteristics are beneficial: a spectrum embedding spreads over the whole spatial extent of the image, and is therefore could be detected at any part and is regardless of pattern size. The invariant mechanism in our work is based on the constant spectrum properties of DCT [10] and epipolar constrain [4]. Details are given in Section 7.2.

1.7.2 Response detection and segmentation: We embed the pattern identification code in DCT domain. Spectrum information in DCT can be obtained by convolution, where tensor computation platforms offer well-optimised performance (PyTorch in this work). Compared to discrete Fourier transform (DFT) [11], DCT is simpler as it abandoned the phase information. This cancels out the shift-invariant property of DFT. To compensate, we deploy the pooling function. Later, a tailored metric measurement is proposed to make the pattern detection more robust. Segmentation of the metric response is carried out by first K-Means clustering and active contour method. Concrete method is given in Section 7.5.

All manipulations are in tensor manner.

1.7.3 Experiments and discussion: We present experiments in both simulation and real-world practise in Section 8.4.

2 Related work
As far as we know, a 3D acquisition system designed particularly to deploy tensor computation has not been seen in published works, therefore hardly related works can be found. This section will describe related works regarding 3D reconstruction to help illustrate the knowledge spectrum.

A general overview of the many 3D acquisition methods proposed thus far is given by Moons et al. [6]. The pro's and con's of structured light methods are also elaborated, of which many versions have been studied; for details see [7, 12, 13].

Two main categories of structured light methods are presented: temporal encoding methods [14–16] and spatial encoding methods [7, 17, 18].

2.1 Pattern generation
Pattern design is an essential part of the structured light approach. The patterns can basically be stripes [19], grids [20] or De Bruijn patterns [21]. White-noise dot patterns have also been reported to provide good performance [12, 15, 22, 23].

2.2 Correspondence search
There is a large body of work studying the correspondence search problem [13, 24–26]. SGM proposed [3] can be considered as the state-of-the-art in corresponding search, along with its variations [27–31]. In [29], authors attribute SGM’s good performance to message passing theory. Due to its individual path-processing, SGM can leverage parallel computation. A work combining SGM stereo matching with a CNN is proposed in [31]. In [27], the correspondence matching path is discussed and innovated further.

2.3 Cooping multiple structured light
Cooping multi-projections is a rational attempt to extend the working range of active illumination based methods. As it is still novel, not much works reported.

Several approaches, which deploy multiple projections simultaneously in one-shot manner, have been proposed [12, 22]. Saggese et al. presents a system using random bandpass white noise pattern [22]. They train a CNN on simulated dataset to learn pattern separation. Multi-projector, multi-camera setting for capturing entire shaped human body is reported in [16].

3 Attenuation measurement
Stereo matching methods perform poorly when directly applied to structured light pairs [5], as the photometrical difference is obvious between what is projected and captured yet projector cannot catch the influence of illumination and reflection on object surface. The obvious photometric difference results in the failure of similarity measurement across the matching image pair. Empirically, a pattern from projector to being captured by camera can be treated as an information loss process. More exactly, its behaviour is similar to a low-pass filter, i.e. the high frequency attenuates fast while low frequency part endures. Our pattern generation belongs to the spectrum-based random patterns. In this case, the information provided by the high-frequency part cannot help correspondence match but hinder it as noise. This renders an attenuation measurement necessary so that the pattern texture is generated only in the proper frequency part; therefore the projector and camera see almost the same. By measuring the spectrum attenuation in a structured light set, the range of spectrum which is suitable for generating pattern textures can be determined. To make the process simple and effective, we devise a spectrum attenuation measurement pipeline.

As DCT [10] is a member of DFT family, it shares the properties of rotation. With the assumption that the attenuation is homogeneous in all directions, measuring the spectrum along any direction would provide a reasonable estimation of attenuation coefficients across spectrum domain. We first propose an invariant mechanism in the spectrum regards epipolar geometry to make the measurement simple.

3.1 Invariant mechanism
The invariant mechanism is based on epipolar geometry and the constant horizontal part of DCT spectrum. Assuming the projector and camera are aligned horizontally (say along x-axis), the constant horizontal spectrum are \( F_{xy} \) with \( x = 0 \). It would be of the appearance as cosine function along y-axis while x component constant as shown in Fig. 3.

The spectrum marked is around the constant horizontal part. As the projector–camera pair establishes an epipolar constrain, epipolar geometry constrains the deformation only along the baseline. In our case, the disparity only changes along the x-axis. It is obvious that, the pattern generated by the constant horizontal
The spectrum remains invariant if the baseline in rectified image plane is horizontal, so does their spectrum. Or to say, the constant spectrum projected is observed the same while captured by the camera on the rectified image plane. For epipolar geometry, readers can refer to [4].

With the constant horizontal spectrum invariance mechanism, we therefore propose our attenuation measurement procedure. At first, one determines the original size of the pattern to be projected, often the native resolution of the projector. We mark the spectrum in the rectified image plane using the constant horizontal spectrum. Then, the pattern is warped back into the projector image plane. After a camera captured the projected pattern. We warp the captured image to the rectified plane and measure the attenuation coefficient. By assuming the very low-frequency part does not attenuate at all, the coefficients of attenuation across different spectrum can be retrieved.

### 4 Pattern generation

The pattern design is important for structured light (and in general active illumination) algorithms.

We vary the original approach that uses a DFT proposed in [15] to a DCT to simplify the parameterisation (i.e. no phase variable in DCT).

In image compression, it has been found that reading the DCT components starting from the DC term and moving in a ‘zig-zag’ pattern towards the opposite corner, allows for efficient compression by truncation [32]. Our ‘band-pass’ random noise patterns follow a similar strategy: we start from a flat spectrum and then put DCT values outside the upper-left triangular region of the DCT domain (i.e. a triangle that has DC as one of its corners) to zero. See the right column in Fig. 4.

### 5 Theory for correspondence search

This section discusses how to adapt census transform to structured light scenario (cf. Section 5.1) from its stereo origin, and a brief review of SGM (cf. Section 5.2).

#### 5.1 Pixel-wise matching metric

In our experiments, the photometrical differences are mainly observed in two ways: (i) projector ‘sees’ the correct structure of pattern, yet without radiometrical interaction with the target surface, i.e. it ‘sees’ identically to what it projects, however the camera captures image as the ensemble effect of illumination and reflection; (ii) projector has smaller depth of field compared to cameras, in other words, projected pattern is prone to blur on the object when distance changes. In experiments the latter process seems like a low-pass filtering. To help correspondence search, the pattern for matching which the camera captured image is firstly blurred by a Gaussian filter (low-pass filtering), in order to bring the projected pattern and captured image to a similar blurry level. Depending on the scene, the optimal size of Gaussian filter varies. In our experiments, a size of 5 works well.

Census transform [9] is a good method here for its well handling on significant illumination change. An example is shown in Fig. 5. It embeds a pixel regarding its neighbourhood as binary vector, therefore robust in illumination change.

#### 5.2 Brief review of SGM

Only pixel-wise cost calculation is often ambiguous and the wrong matches can have a lower cost than the correct ones, while a global optimisation is an NP-complete problem [33] and unsolvable in tolerable time, henceforth SGM establishes image correspondence by approximating a global minimised matching cost.
The idea of SGM is to perform path-wise optimisation along a few 1D paths to approximate the global cost (as shown in (2)). In SGM, cost function in (1) is broken down into path-wise optimisation

\[ E(D) = \sum_p (C(p, D_p)) + \sum_{q \in \mathbb{N}_p} P_q T \left| D_p - D_q \right| = 1 \]

(1)

Let us assume \( X \) is a \( N \)-dimensional tensor, using i.e. \( i = [i, i, \ldots, i] \) \( \in \text{idx}(X) \) to index the entries in all dimensions. \( \text{idx}(X) \) are all the indexes available for tensor \( X \). If given a shift vector \( s = [s, s, \ldots, s] \), \( X' = \ominus(X, s) \) will retrieve the entries \( X \) shifted by \( s \) and return \( X' \), which satisfies

\[ vi \in \text{idx}(X') \Leftrightarrow vi - s \in \text{idx}(X), \quad X'(i') = X(i - s) \]

(3)

One shall pay attention that, ‘tensor shift’ consists of a re-indexing in a tensor instead of forming a new one, which prevents the effort of allocating new memory. Python-style indexing grammar is used to help illustration.

### 6.2 Census transform in tensor form

Let us consider one channel of census transform, say a neighbour with shift \( i \) regarding the kernel centre pixel for all pixels in the image. Then the response of this channel in census transform is calculated as \( I(p - i) \). Obviously, this could be naturally parallelised in tensor form, as shifting the same distance for all the pixels is equivalent to shifting the whole tensor, as described in (3). In other words, extracting one channel of census transform for the whole image accomplishes in a single tensor manipulation (Fig. 6).

\[ \mathcal{I} = \text{CensusTransform}(I, d_{min}, d_{max}) \]

\[ \mathcal{I} = \text{CensusTransform}(I, d_{min}, d_{max}) \]

(6.3 Tensor-SGM)

Remind that the core processing formula in SGM is as (2). To make the appearance consistent to tensor manipulation in practice, e.g. Python, we use \( L(r, d, h, w) \) to describe the same information, where \( h \) and \( w \) are indexes of height and width. There are three basic types of scanning paths in SGM: horizontal, vertical and diagonal. As one might notice from (2), the less easy part is how to process the terms inside \( \min \) function, concerning both path manipulation \( p - r \) and disparity manipulation \( d + 1, d - 1 \) using ‘tensor shift’.

### 6.3.1 Access spatial neighbourhood using ‘tensor shift’

We begin from horizontal paths, where the previous neighbour of each pixel is next to its right/left, in other words, for each pixel related cost in \( L(r, d, \ldots, j) \), its corresponding previous pixel(s) to the right/left can be retrieved as \( L(r, d, \ldots, j + 1) \) or \( L(r, d, \ldots, j - 1) \), respectively.

Similarly, for vertical paths, to access corresponding previous step pixel of each pixel \( L(r, d, i, \ldots) \) would be simply \( L(r, d, i + 1, \ldots) \) or \( L(r, d, i - 1, \ldots) \), as illustrated in Fig. 8.

For diagonal paths, depending on how one sweeps across the image, i.e. in vertical manner or horizontal manner, slightly different formulas could be derived. We show an example here the vertical case as shown in Fig. 8b. The previous pixels of \( L(r, d, i, 2:) \) with one step back are \( L(r, d, i - 1:) \). Pay attention that there is one pixel without \( p - r \) neighbour (the red pixel), so that to avoid illegal memory access.

### 6.3.2 Access depth neighbourhood using ‘tensor shift’

As shown in the first sub-figure in Fig. 8d, (-1) \(-\rightarrow-\) \(-\rightarrow-\) in disparity channel would extract the cost of identical disparity for each pixel. To describe disparity change \( d + 1 \) and \( d - 1 \), as shown in the
second and third sub-figures in Fig. 8d, a simple shift would do. $L_0(p - r, d - 1)$ could be retrieved using $(:) ← (2:)$ in disparity index. Similarly, $L_0(p - r, d + 1)$ could be collected as $(:) ← (: - 2)$ in disparity index. Again, notice that one pixel is without legal access.

For $\min(L_0(p - r, i)) + P_2$, situation is a bit different, since a $\min$ function changes the size of the retrieved tensor. By assigning the minimum values from previous step $(:) ← \min(:)$ as the last sub-figure in Fig. 8d, one should reshape the tensor after $\min$ function so that it can be broadcast.

6.3.3 Combine both: tensor form: By combining the manipulation in Sections 6.3.1 and 6.3.2, we could write down the formula of diagonal paths as Fig. 8f for instance. Notice that tensor shifts take place in both spatial dimension(s) and disparity dimension (Fig. 9).

Vertical paths and horizontal paths could be similarly derived by cancelling the spatial shift term (i.e. $:-2$). By appending a diagonal path after a vertical/horizontal path inter-changing at every row/column, another eight paths could be generated as Fig. 8c, so that the suggested 16 paths could be achieved.

7 Beyond: framework to coop multiple structured light

The invariant mechanism and tensorisation leads to a framework which coops multiple structured lights to work simultaneously [5]. It can support applications which require a 3D reconstruction from multiple views, e.g. face identification for online bank service. This section aims to present our works excluded to the one-shot system but the same framework, i.e. tensorisation, as an introductory to show what else our proposed tensorisation framework can produce. To avoid confusion to the contents of the tensorised one-shot structured light, we leave the details in the original paper.

The framework allows multiple structured lights working simultaneously with an acquisition strategy. A projected pattern in combination with a camera allows for a structured light approach. This is beneficial, given the weakly textured surfaces we are dealing with. Yet, where projection patterns overlap, the system automatically changes over to a multi-view approach. In order to let the system automatically detect whether a single projection versus an overlap is observed, we watermark the different projection patterns while preserving enough textures for correspondence match. In the reported system, two projectors and two cameras are deployed. Each camera–projection pair consists of a one-shot structured light set and the two cameras consist of a multi-view stereo. An example of pattern identification with two patterns’ presence is shown in Fig. 10 and the results in Fig. 11.

7.1 Base pattern generation revise

Assuming the projector has the native resolution of $p_x × p_y$ where $p_x$ is the number of rows and $p_y$ the columns, the pattern to generate in the rectified plane should be accordingly to the camera which observes the pattern, i.e. corresponding camera–projector epipolar pair. Say the left projector (projector$_x$) is observed by the right camera (camera$_y$) with homography $H_{xy}$ to warp the pattern to the rectified image plane, the base pattern is generated in the rectified image plane warped by $H_{xy}$. After the base pattern is generated, a post-processing step is performed. As the pattern is generated from spectrum, it is not straight-forward to evaluate the peaks and valleys in spatial domain. A few very high or low values can compress the contrast of pattern significantly. We use a histogram post-processing to normalise the pattern first, before sending to embedding. The post-processing method is described in Section 7.4. As suggested in [5], the spectrum across different image size can be cast through wavelength. Therefore, the identical spectrum can be determined under the size change of pattern/window.

7.2 Constant spectrum embedding

After the base pattern is generated, we convert the pattern into its spectrum. We define the code of each pattern by $[k, N]$ [5]. The two parameters define the wavelength of each cosine component in DCT domain where $k$ is the spectrum to be embedded in a full image size of $N$. According to formula of standard DCT transform (so called DCT-II), the two parameters are enough to define the wavelength. Fig. 12a shows the spectrum to embed. Since DCT ignored the phase information compared to DFT, the phase change affects the response (high values only detected at the peaks). The handling of phase change is presented in Section 7.5 by pooling.

7.3 Patterns fusion

We generate and embed the pattern in rectified plane first and then use $H_{xy}^r$ to warp the generated pattern back (the other one using $H_{xy}^l$). So that the projected pattern can deploy the full range of the projector pixels without interpolation vacuum. Then the same post-process is applied. In this way, the embedded code/spectrum can be detected in the captured image, by warping to the rectified image plane as well. Figs. 13a and c show the generated and embedded patterns in the two rectified image planes for one projector. After the generation, a post-processing is applied, and then warped back to the projector image plane. One can notice that (i) the rotation angle is different and (ii) the warped back patterns are with better contrast. At last, we fuse the two patterns together to get the final
patterns are warped back to projector image plane, then generate the base patterns.

Fig. 12 The pipeline: Spectrum embedding in rectification planes
(a) Warping to rectified image plane and determine corresponding size, then generate the base patterns, (b) Embedding in the spectrum, (c) Patterns generated in spatial domain, (d) Empty pattern in projector plane, (f) Warp the generated patterns to projector plane. (a) Fuse the generated patterns to get the projector pattern.

Fig. 13 Patterns embedded in rectified plane warped back and fuse into one pattern
(a), (c) Generated embedded pattern in rectified image planes, (b), (d) Embedded patterns are warped back to projector image plane, (e) Fused pattern.

Fig. 14 Left pattern is directly generated from random spectrum while the right is the same pattern after post-processing.

Fig. 15 Spectrum response examples
(a) Averaged kernel by summing up all detected kernels together, (b) Random sampled kernels.

Fig. 16 Procedure to extract metric response of embedded spectrum

7.4 Pattern post-processing
The same post-processing step is used several times in our pipeline. As we perform both the base pattern generation and spectrum embedding in the frequency domain. The spatial peaks, both low and high, generated from spectrum is not easy to control and can reduce the pattern contrast significantly, which renders a post-processing necessary. We first apply a histogram thresholding, removes the top and low pixels with given percentage. Afterwards, a standard normalisation that converts the value range to [0, 1] is performed. The parametrization one shall experiment by oneself, in our case, 20% seems working well. As shown in Fig. 14, after the post-processing the pattern gains a much better contrast. Also, histogram manipulation would not remove the embedding.

7.5 Pattern response detection
Spectrum manipulation is suitable for tensor computing. With nowadays tensor processing frameworks, fast processing is easy to reach. Our response detection method contains mainly: convolution, pooling and metric computation.

As widely known, DFT is a convolutional process, so is DCT. For each spectrum we embed in the rectified image plane, we simply generate the corresponding convolutional kernel which extracts DCT response for a certain spectrum. Theoretically, for one embedding, only one kernel would be enough. However, during the experiment, we found a phenomenon that the embedded spectrum can ‘drift’, as shown in both Figs. 15a and b. Though we only embedded one spectrum, yet during the response simulation test, the neighbourhood can also show strong response. Statistically, we found that in our system the strong response distributed mostly in the distance-1 pixel neighborhood. Hence, for each embedded spectrum we have six convolution kernels to evaluate, i.e. both the embedded spectrum and its five neighbours.

As mentioned previously, DCT does not contain the phase information. As a result, the strong response can only be detected around the peak, which is against our intention to detect the illuminated region of the pattern. To compensate, we append an pooling process after the convolution. It is simple to propose that the pooling window size is the wavelength of the spectrum for each convolutional kernel corresponding to.

By evaluating the neighbourhood, we are able to detect the ‘drifted’ spectrum, however, it also raises the question that how can we put the information into a measurement which tells the system how likely the pattern is detected. We propose a metric to remedy it. We name it exponential sum metric.

\[ m = \sum_{r \in N} n \times e^{\alpha r} \]  

where \( r \) is the pooling response across the detection pixels neighbourhood and \( \alpha \) is a scale parameter (0.1 in our system). The metric can reach high values while one or a few response in the neighbourhood is high, while the low values are compressed.

To summarise, as shown in Fig. 16, we first generate the convolutional kernels regarding the embedded spectrum and its neighbourhood, then we apply these kernels in the convolution layer to retrieve the convolutional response. Later, a pooling layer is applied to compensate the loss of phase information. At last, exponential sum metric is applied to integrate the information provided by the embedded spectrum and its neighbourhood. The integrated information is the input for the next process: response segmentation to identify which part is lit by which projector.

7.6 Easy method for response segmentation
After the metric response is extracted, we segment the response to identify the area illuminated by each pattern. Figs. 17a–d present the procedures in last subsection, from warping to rectified plane to metric response extraction. To segment the metric response, we first apply K-Means clustering to the metric response. The number pattern. To illustrate, Fig. 13 is zoomed in so that details can be seen.
8.1 Innovation using tSGM framework

8.1.1 Variant parametrisation: Originally in SGM only two parameters are required: P1 and P2. In [30], they are extended to 20 parameters for a 4-path SGM. For each path, five parameters are provided: \( P1(r), P2(r), w(r), P1(r)^{\prime}, P2(r)^{\prime} \) path dependent smoothness regulators. \( P1(r) \) and \( P2(r) \) act instead of \( P1(r) \) and \( P2(r) \) once the intensity gradient at current pixel is beyond certain threshold, so to adapt to sudden intensity changes, e.g. boundaries. Obviously, \( P1(r), P2(r) \) and \( w(r) \) could be parametrised in tSGM easily by assigning them accordingly to each path. The implementation of \( P1(r), P2(r) \) could be naturally achieved by applying sobel filtering to the image, and then threshold for an index on gradient amplitude. By applying this index, \( P1(r), P2(r) \) can be directly applied.

8.1.2 Tensor-MGM: We demonstrate the ease of tuning paths by implementing MGM proposed in [27] in tensor form (tMGM), with results shown in Fig. 18. MGM improves SGM by coop message passing between neighbour search paths. It is achieved by integrating cost information from an orthogonal path at every pixel as shown in the following equation:

\[
L_d(p, d) = C(p, d) + \sum_{x \in \{r, f\}} \frac{1}{\pi} \min_{d' \in D} (L_r(p - x, d') + V(d, d'))
\]

8.2 Reconstruction results

Our setup uses a 1280 × 800 projector, a camera with original resolution 2448 × 2048, which is downscaled to 1224 × 1024 in order to balance pixel–pixel ratio between projector and camera. The setup is calibrated beforehand. A good calibration library is [35], Rectification homography from calibration alone might not be accurate enough as calibration errors accumulate, while SGM is prone to epipolar line [36]. Some additional rectification methods might be in need, such as [37, 38]. In our experiments, surfaces are meshed using scale-space surface reconstruction from triangulated point cloud proposed in [39]. Disparity map is filtered using bilateral filtering [40].

The first row contains the captured images from beholding camera with target object being illuminated by projected pattern, while the bottom row is the reconstructed surface. For SGM used in all sets, P1 is always chosen to be 8. P2 varies from 32 to 256 that depend on whether one wants more smoothness or preserving more discontinuities. See more details and feedback through our released code.

On multiple planes: Figs. 19a and f show the reconstruction of paper boxes, mainly planes with sharp corners and discontinuities. Our proposed approach reconstructed the planes well, with the preservation of the corners.

On curve surface: Figs. 19b and g show the reconstruction of a ball surface. The centre part and the part close to boundary where illumination and the structure of pattern still survive are well reconstructed. While approaching the boundary, the reconstruction weakens. As a suspect, the defects on the boundary may be due to calibration errors, which can accumulate and are usually larger at the boundary of the camera rather than the centre [36], since SGM...
is an epipolar sensitive algorithm. Slight calibration errors might defect the correspondence searching quality significantly.

On human skin: Reconstructions on human skin, which diffuse projected pattern more severely than paper and bust, are presented in Figs. 19c and h and d and i, on foot and hand, respectively. As on the skin the patterns are more blurred compared to on paper and plaster, the reconstruction results appears to be noisier.

On bust: Figs. 19e and j are the reconstruction results of a head bust, which has good preservation of the projected pattern in the term of blurring yet complicated shape (one can refer to the colour image in Fig. 1a for the imaged bust). In the reconstruction shown, P1 is 8 and P2 is 128. One may notice that the details (e.g. eyes, lips, hair etc.) are properly preserved, while rather flat parts (e.g. forehead, cheeks) appear to be smooth.

8.3 4D capturing

Being an ‘one-shot’ approach, our method is naturally able to capture 4D sequences, i.e. 3D reconstruction + time (Fig. 20).

Using 3D reconstruction opens new possibilities such as in sport analysis [41]. As a single frame of the captured image is enough to yield its corresponding 3D reconstruction, and the projected pattern is consistent, we can apply our reconstruction algorithm to a video capture. A video of 4D capturing example can be found: https://www.bilibili.com/video/av55226158.

8.4 Beyond: cooping multiple structured light

In this section, we report our experiment results of the proposed framework. We carry out the experiment in two stages: (i) simulation and (ii) practice on real face.

8.4.1 Simulation: We first perform the response detection on simulation. After both patterns are generated (Figs. 21a and b), we warp them to the rectified image planes accordingly. As each pattern are observed by two cameras, and each projector–camera pair constitutes an epipolar pair, there are four warpings needed (two for each), as shown in (a1, a2, b1, b2). Further, we carry out the response detection procedures as described in Section 7.5. The response results are shown in the right column of Fig. 21(a3, a4, b3, b4). One can observe that the embedded response can be detected robustly.

In Fig. 15, we visualise the response in the perspective of kernels, i.e. left: averaged kernel, right: random sampling of detected spectrum. We notice that during the projection-capture
process, the embedded spectrum seems to drift to the neighbourhood. We further investigate and found that in our system the drift distributed in the distance-1 neighbourhood statistically. As described previously in Section 7.5, we use the embedded spectrum together with its neighbourhood to perform the spectrum detection, i.e. convolutional kernels.

### 8.4.2 On real surface:

Then we move on to experiment on human face. As face has complicated shape, the projected pattern is deformed by projective geometry while being captured by the camera, also with reflection changes. These subsidiary factors render the response on real surface less strong than simulation. Yet, it is still strong enough, as shown in Figs. 22 and 23.

Figs. 22 and 23a show the original capture. It is then warped to rectified image plane as shown in Figs. 22 and 23b, where we perform the response detection as it is the rectified image plane where the code is embedded. Figs. 22 and 23c show the averaged spectrum, as also observed in the simulation experiment, spectrum drift also takes place. One can tell from the two neighbour peaks around our originally embedded spectrum. Then convolution is performed using the averaged spectrum kernel and its neighbourhood kernels. Wavelength pooling follows to compensate the lacking of phase information of DCT transformation and then we calculate metric response, which is visualised in Figs. 22 and 23d. Figs. 22 and 23e show the segmentation result of detected area using active contour. We notice that on some boundary parts, some false negatives are made. We suppose this is because of the convolution window nature. As our spectrum is extracted by convolution, say by window size \([W, W]\), then on some boundaries, the response decay due to the fact that only part of the window is lit by embedded pattern. These decayed response area is identified as negative in the K-means clustering step. Further processing is in need if the boundaries here are of concern.

The segmentation result of the right pattern on face is shown in Fig. 23. The sequence of detection process is identical to the left pattern detection. In Fig. 23e, which shows the averaged spectrum, the spectrum ‘drift’ seems less severe. Yet, this might be an illusion as the embedded spectrum in the right pattern is at higher frequency, where the signal decays more severe. From the convolution image sequence, one can still observe the strong response spectrum distributed into the neighbourhood of the original embedded spectrum.

At last, by integrating the segmentation results, we show the detection results in Fig. 24. The red part is sent to left structured-light pair for reconstruction, while the green part to the right pair. The yellow part, which indicates two patterns overlapped, is reconstructed by multi-view stereo, using both the cameras in the system.

### 9 Conclusions

In this paper, we build a novel and practical one-shot structured light 3D acquisition system fully designed and implemented in tensor computation. We begin with a finding that ‘tensor shift’ manoeuvre can access neighbourhood at the full-image scale with ease. Based on this finding, we designed and implemented the reconstruction system and reconstructed 3D/4D objects in practice. The high quality and efficiency convince us that tensor computation can lead to more innovations (e.g. random field alike algorithms).

We propose a watermarking pattern framework to allow multiple structured light systems working together. As the invariance mechanism is supported by both the spectrum properties and epipolar geometry, we consider our method is theoretically profound, which shall lead to reliable practice. Also, our design carries out the computation using the tensor computation where parallel computation can be easily tamed. At last, to tackle the spectrum drift, our proposed metric manages to produce reliable results. Yet, our coding strategy is still native, further investigation into the maximum payload might lead to more applications. Nevertheless, the segmentation method we deploy can work robustly yet not well tailored. We glimpse that a more suitable segmentation algorithm shall be investigated for this certain kind of image, i.e. spectrum response.

### 10 References

[1] Pytorch: http://pytorch.org, accessed: September 2019
[2] Cambricon: http://www.cambricon.com, accessed: April 2020
[3] Hirschmuller, H.: ‘Accurate and efficient stereo processing by semi-global matching and mutual information’. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, 2005. CVPR 2005, San Diego, USA, 2005, vol. 2, pp. 807–814
[4] Hartley, R., Zisserman, A.: ‘Multiple view geometry in computer vision’. (Cambridge University Press, Cambridge, UK, 2003)
[5] Chen, X., Van Gool, L.: ‘Watermarking structured light patterns for one-shot, extendable 3D scanning’. Eleventh Int. Conf. on Information Optics and Photonics (CIOP) 2019, Xi‘an, China, 2019, http://doi.org/10.1117/12.2549975

This is an open access article published by the IET in partnership with Shenzhen University under the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0/).
Moons, T., Van Gool, L., Vergauwen, M.: ‘3d reconstruction from multiple images’. *Found. Trends Comput. Vis.*, 2009, 4, (4), pp. 287–404

Sagawa, R., Furukawa, R., Kawasugi, H.: ‘Dense 3d reconstruction from high frame-rate video using a static grid pattern’. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2014, 36, (9), pp. 1733–1747

Chen, X., Van Gool, L.: ‘Fully in tensor computation manner: a one-shot dense 3D acquisition method using structured light’. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2016, 38, (2), pp. 390–404

Ryan Fanello, S., Rhemann, C., Tankovich, V.: ‘Learning-based feature extraction for active 3d scan with reducing color crosstalk of multiple pattern projections’. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2017, 39, (7), pp. 1310–1324

Zabih, R., Woodfill, J.: ‘Non-parametric local transforms for computing visual correspondence’. European Conf. On Computer Vision, Stockholm, Sweden, 1994, pp. 151–158

Ahmed, N., Natarajan, T., Rao, K.R.: ‘Discrete cosine transform’. *IEEE Trans. Comput.*, 1974, 100, (1), pp. 90–93

Oppenheim, A.V.: ‘Discrete-time signal processing’ (Pearson Higher Ed, USA, 2011, 3rd edn.)

Vo, M., Sarafianos, S.G., Sheikh, Y.: ‘Texture illumination separation for single-shot structured light reconstruction’. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2016, 38, (2), pp. 390–404

Ryan Fanello, S., Rhemann, C., Tankovich, V. et al.: ‘Hyperdepth: learning depth from structured light without matching’. Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, Las Vegas, USA, 2016, pp. 5441–5450

Gupta, M., Nayar, S.K.: ‘Micro phase shifting’. 2012 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 2012, pp. 813–820

Couture, V., Martin, N., Roy, S.: ‘Unstructured light scanning to overcome interreflections’. 2011 IEEE Int. Conf. on Computer Vision (ICCV), Barcelona, Spain, 2011, pp. 1895–1902

Petrovic, T., Pribanic, T., Donlic, M., et al.: ‘Efficient separation between projected patterns for multiple projector 3d people scanning’. IEEE Int. Conf. on Computer Vision Workshops (ICCV Workshops), Venice, 2017

Kawasaki, H., Furukawa, R., Sagawa, R. et al.: ‘Dynamic scene shape reconstructions from a single structured light pattern’. IEEE Conf. on Computer Vision and Pattern Recognition, 2008. CVPR 2008, Anchorage, Alaska, USA, 2008, pp. 1–8

Zbontar, J., Calakli, F., Taubin, G.: ‘One-shot scanning using de bruijn spaced grids’. 2009 IEEE 12th Int. Conf. on Computer Vision Workshops (ICCV Workshops), Kyoto, Japan, 2009, pp. 1786–1792

Boley, K.L., Kak, A.C.: ‘Color-encoded structured light for rapid active rangings’. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1987, PAMI-9, (1), pp. 14–28

Proesmans, M., Van Gool, L., Oosterlinck, A.: ‘One-shot active 3D shape acquisition’. Proc. of the 13th Int. Conf. on Pattern Recognition, 1996, Vienna, Austria, 1996, vol. 3, pp. 336–340

Vuiylsteke, P., Oosterlinck, A.: ‘Range image acquisition with a single binary-encoded light pattern’. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1990, 12, (2), pp. 148–164

Sagawa, R., Furukawa, R., Matsumoto, A., et al.: ‘Learning-based feature extraction for active 3d scan with reducing color crosstalk of multiple pattern projections’. 2017 IEEE Int. Conf. on Robotics and Automation (ICRA), Marina Bay Sands, Singapore, 2017, pp. 5074–5080

Martinez, M., Stiefelhagen, R.: ‘Kinect unleashed: getting control over high resolution depth maps’. Machine Vision Applications, Kyoto, Japan, 2013, pp. 247–250

Sinha, S.N., Scharstein, D., Szeliski, R.: ‘Efficient high-resolution stereo matching using local plane sweeps’. Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, Columbus, Ohio, USA, 2014, pp. 1582–1589

Seki, A., Pollefeys, M.: ‘Patch-based confidence prediction for dense disparity map’. British Machine Vision Conf., York, UK, 2016

Bleyer, M., Rhemann, C., Rother, C.: ‘Patchmatch stereo-stereo matching with slanted support windows’. British Machine Vision Conf., Dundee, UK, 2011, vol. 11, pp. 1–11

Facciolo, G., De Franchis, C., Meinhardt, E.: ‘MGM: A significantly more global matching for stereovision’. British Machine Vision Conf. 2015, Swansea, UK, 2015

Spangenberg, R., Langner, T., Rojas, R.: ‘Weighted semi-global matching and center-symmetric census transform for robust driver assistance’. Int. Conf. on Computer Analysis of Images and Patterns, York, UK, 2013, pp. 34–41

Droey, A., Haubold, C., Avidan, S., et al.: ‘Semi-global matching: a principled derivation in terms of message passing’. German Conf. on Pattern Recognition, Münster, Germany, 2014, pp. 43–53

Michael, M., Salmen, J., Stallkamp, J., et al.: ‘Real-time stereo vision: optimizing semi-global matching’. Intelligent Vehicles Symp. (IV), 2013 IEEE, Gold Coast, Australia, 2013, pp. 1197–1202

Zbontar, J., LeCun, Y.: ‘Stereo matching by training a convolutional neural network to compare image patches’, *J. Mach. Learn. Res.*, 2016, 17, (1–2), p. 2.

Pennebaker, W.B., Mitchell, J.L.: ‘JPEG: still image data compression standard’ (Springer Science and Business Media, Berlin, Germany, 1992)

Boykov, Y., Veksler, O., Zabih, R.: ‘Fast approximate energy minimization via graph cuts’. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2001, 23, (11), pp. 1222–1239

Kass, M., Witkin, A., Terzopoulos, D.: ‘Snakes: active contour models’. *Int. J. Comput. Vis.*, 1988, 1, (4), pp. 321–331

Moreno, D., Taubin, G.: ‘Simple, accurate, and robust projector-camera calibration’. 2012 Second Int. Conf. on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), Zürich, Switzerland, 2012, pp. 464–471

Hirschmuller, H., Gehrig, S.: ‘Stereo matching in the presence of sub-pixel calibration errors’. CVPR 2009. IEEE Conf. on Computer Vision and Pattern Recognition, Miami, FL, USA, 2009, pp. 437–444

Monasse, P.: ‘Quasi-Euclidean epipolar rectification’, *Image Process. On Line*, 2011, 1, pp. 187–199

Harley, R.I.: ‘Theory and practice of projective rectification’, *Int. J. Comput. Vis.*, 1999, 35, (2), pp. 115–127

Digne, J., Morel, J.M., Souzani, C.M., et al.: ‘Scale space meshing of raw data point sets’, in ‘Computer graphics forum’ (Wiley Online Library, Hoboken, NJ, USA, 2011), vol. 30, pp. 1630–1642

Facciolo, G., Oosterlinck, A.: ‘Semi-global matching: a principled derivation in terms of message passing’. 2016 IEEE, Gold Coast, Australia, 2013, pp. 1197–1202

Ito, K., Maeda, K., Fujiwara, I., et al.: ‘Time change in strain distribution on the lateral and plantar surfaces of the foot during walking’, *Foot. Ankle. Surg.*, 2016, 2, (22), p. 114