Supplemental Document

Optics EXPRESS

Synchronization-free top-down illumination photometric stereo imaging using light-emitting diodes and a mobile device: supplement

EMMA LE FRANCOIS,[1,*] JOHANNES HERRNSDORF,[1] JONATHAN J. D. MCKENDRY,[1] LAURENCE BROADBENT,[2] GLYNN WRIGHT,[2] MARTIN D. DAWSON,[1] AND MICHAEL J. STRAIN[1]

[1] *Institute of Photonics, Department of Physics, University of Strathclyde, Glasgow G1 1RD, UK*
[2] *Aralia Systems, Bristol Robotics Laboratory, Bristol BS16 1QY, UK*
[*] *emma.le-francois@strath.ac.uk*

Supplement DOI: https://doi.org/10.6084/m9.figshare.13298588

Parent Article DOI: https://doi.org/10.1364/OE.408658

Synchronization-Free Top-Down Illumination Photometric Stereo Imaging Using Light-Emitting Diodes and a Mobile Device: supplemental document

This document provides supplementary information to "Synchronization-Free Top-Down Illumination Photometric Stereo Imaging Using Light-Emitting Diodes and a Mobile Device". First of all, the document details the demodulation process for the recovery of the four images of each LED. Then, in order to assess our Fast Marching algorithm, we provide a comparison with Yvain Queau dataset. Finally, to benchmark our top-down illumination method, we demonstrate that our imaging technique provides similar results and errors as current works in the literature.

1. MEB-FDMA DECODING ALGORITHM

In conventional FDMA, decoding is achieved by a Fourier transform where for each frequency ν_i, there are two real-valued orthogonal base vectors, namely $\sin(2\pi\nu_i t)$ and $\cos(2\pi\nu_i t)$. Note that these two FDMA base vectors are phase shifted by $\pi/2$ to each other. Decoding of an MEB-FDMA signal is analogously done via a set of orthogonal base vectors resulting from phase shifts of the original transmitted $s_{i,j}$. The received signal $r^{(i)}$ from the i^{th} emitter at a given receiver (camera pixel) will have a certain intensity I_i and it will be phase shifted by a phase $\Delta_j = k + \alpha$ (compare Eq. (1) in the main text):

$$r_j^{(i)} = I_i \left((1-\alpha) s_{i,1+(j-1+k)\%n} + \alpha s_{i,1+(j+k)\%n} \right) \tag{S1}$$

Eq. (S1) assumes that the intensity is integrated during the sample time, which is the case in our experiment. The actual received signal r is then the sum of the contributions from all emitters:

$$r_j = \sum_{i=1}^{N} r_j^{(i)} \tag{S2}$$

The task of the decoding algorithm is to calculate the unknown I_i from the known r.

For a given i we construct an $n \times n$ matrix $S_{k,j}^{(i)}$:

$$S_{k,j}^{(i)} = s_{i,1+(j-1+k)\%n} \tag{S3}$$

where s is given by Eq. (7) in the main text. S does not have full rank:

$$l_i = \operatorname{rank}(S^{(i)}) = 2^i < n \tag{S4}$$

Its rows span the subspace of $\mathbb{R}^n$ that contains all phase-shifted versions of $s_{i,j}$, but they are not orthogonal. We therefore use an orthonormalization algorithm "GS" to construct $l_i \times n$ matrices $D^{(i)}$:

$$D^{(i)} = \mathrm{GS}(S^{(i)}) \tag{S5}$$

In this work, GS is the stabilized Gram-Schmidt algorithm, but other orthonormalization procedures can be used as well. The orthogonal decoding matrices $D^{(i)}$ can be used to decompose r into its orthogonal components c_k:

$$c_k^{(i)} = \sum_{j=1}^{n} D_{k,j}^{(i)} r_j \tag{S6}$$

In conventional FDMA, the signal I_i would be the amplitude of the vector $c^{(i)}$. However, due to the non-orthogonality of the process in Eq. (S1), a slightly different approach needs to be taken in

MEB-FDMA. In a first step, we reconstruct $r^{(i)}$:

$$r_j^{(i)} = \sum_{k=1}^{l_i} c_k^{(i)} D_{k,j}^{(i)} \tag{S7}$$

Then we note that each of the sequences $s_{i,j}$ has at least one occasion where two 1's are transmitted directly after each other. This means that $r^{(i)}$ will have at least one entry with the intensity corresponding to a synchronized 1-level, and this intensity will be the maximum of $r^{(i)}$. Therefore we have:

$$I_i = \max(r^{(i)}) \tag{S8}$$

Eqs. (S3) to (S8) are the decoding algorithm for MEB-FDMA. The orthonormalization process in Eq. (S5) is computationally demanding. However it needs to be carried out only once upon system manufacture/installation and then the matrices $D^{(i)}$ can be stored permanently in a memory on the receiving clients.

A. Illustration on the example of two emitters

Consider two emitters, $N = 2$. Then according to Eq. (6) in the main text we start off with two simple square waves:

$$s_{i,j}^{(0)} = \begin{bmatrix} -1 & 1 & -1 & 1 \\ -1 & -1 & 1 & 1 \end{bmatrix} \tag{S9}$$

Then the Manchester encoded transmitter signal s is, according to Eq. (7) in the main text:

$$s_{i,j} = \begin{bmatrix} 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \end{bmatrix} \tag{S10}$$

The decoding matrices $D^{(i)}$ are calculated by Eqs. (S3) and (S5) to:

$$D^{(1)} = \begin{bmatrix} 0.35 & -0.35 & -0.35 & 0.35 \\ 0.35 & 0.35 & -0.35 & -0.35 \\ 0.35 & -0.35 & -0.35 & 0.35 \\ 0.35 & 0.35 & -0.35 & -0.35 \end{bmatrix} \tag{S11}$$

$$D^{(2)} = \begin{bmatrix} 0.35 & -0.35 & 0.35 & -0.35 \\ 0.61 & 0.20 & -0.20 & 0.20 \\ 0 & 0.58 & 0.29 & -0.29 \\ 0 & 0 & 0.5 & 0.5 \\ -0.35 & 0.35 & -0.35 & 0.35 \\ -0.61 & -0.20 & 0.20 & -0.20 \\ 0 & -0.58 & -0.29 & 0.29 \\ 0 & 0 & -0.5 & -0.5 \end{bmatrix} \tag{S12}$$

The numbers in Eqs. (S11) and (S12) are not given with full numerical precision.

Let us now look at a specific example, where $I_1 = 1$ and $I_2 = 2$, and emitter 1 has a phase-shift of 0.3 and emitter 2 a phase shift of 1.7. Then the received signal r is calculated by Eqs. (S1) and (S2):

$$r = \begin{bmatrix} 0.2 & 1.6 & -0.2 & -0.4 & 1.8 & -2.4 & -1.8 & 1.2 \end{bmatrix} \tag{S13}$$

Note that an arbitrary DC offset can be added to r without compromising the evaluation below. The orthogonal components of r according to Eq. (S6) are:

$$c^{(1)} = \begin{bmatrix} 1.98 & 0.85 \end{bmatrix} \tag{S14}$$

$$c^{(2)} = \begin{bmatrix} -0.85 & -0.82 & 3.23 & 0 \end{bmatrix} \tag{S15}$$

The contributions of each emitter to r are then calculated by Eq. (S7):

$$r^{(1)} = \begin{bmatrix} 1 & -0.4 & -1 & 0.4 & 1 & -0.4 & -1 & 0.4 \end{bmatrix} \tag{S16}$$

$$r^{(2)} = \begin{bmatrix} -0.8 & 2 & 0.8 & -0.8 & 0.8 & -2 & -0.8 & 0.8 \end{bmatrix} \tag{S17}$$

Then finally we correctly decode with Eq. (S8):

$$I_1 = \max(r^{(1)}) = 1 \tag{S18}$$

$$I_2 = \max(r^{(2)}) = 2 \tag{S19}$$

2. PROOFS

A. Proof of equivalence of Eqs. (1) and (2) in the main text

By setting $\alpha = 0$, (1)⇒(2) in the main text is directly shown. For the other direction, we reorder the left hand side of Eq. (1) in the main text:

$$\begin{aligned} &\sum_{j=1}^{n} s_{i,j} \left((1-\alpha) s_{i',1+(j-1+k)\%n} + \alpha s_{i',1+(j+k)\%n} \right) \\ =&(1-\alpha) \sum_{j=1}^{n} s_{i,j} s_{i',1+(j-1+k)\%n} \\ &+ \alpha \sum_{j=1}^{n} s_{i,j} s_{i',1+(j+k)\%n} \end{aligned} \tag{S20}$$

If Eq. (2) in the main text is fulfilled, then the right hand side of Eq. (S20) is 0 and thus (1)⇐(2) in main the text is shown.

B. Proof of Eq. (S4)

Lemma 1. Given a row vector $s \in \mathbb{R}^n$, define $k_0 \in \{1, \dots, n\}$ via Eq. (S21):

$$\begin{aligned} k_0 = \min \{ k \in \{1, \dots, n\} \,|\, &\exists \beta \in \mathbb{R}, \forall i = 1, \dots, n : \\ &s_{1+(i-1+k)\%n} = \beta s_i \} \end{aligned} \tag{S21}$$

Then:

1. $\beta \in \{-1, 1\}$
2. s is periodic. If $\beta = 1$ then the period is k_0 and if $\beta = -1$ then the period is $2k_0$
3. If a matrix S is constructed from s using Eq. (S3), then its rank is $\text{rank}(S) \leq k_0$

Proof. We calculate the modulus of s:

$$\begin{aligned} |s|^2 = \sum_{i=0}^{n} s_i^2 = \sum_{i=0}^{n} s_{1+(i-1+k_0)\%n}^2 = \beta^2 |s|^2 \\ \Rightarrow |\beta| = 1 \end{aligned} \tag{S22}$$

The periodicity of s follows directly from Eq. (S22).

The periodicity of s directly implies that $\text{rank}(S) \leq k_0$, because all the rows from k_0 on are periodic repeats of the rows 1 to k_0. □

Now let $k = k_r + q2^i, \quad k_r < 2^i, q \in \mathbb{N}$. Then Eq. (7) in the main text gives:

$$s_{i,j+k} = (-1)^q s_{i,j+k_r} \tag{S23}$$

The special case of $k_r = 0$ implies directly that in Lemma 1:

$$k_0 \leq 2^i \tag{S24}$$

and therefore following Lemma 1:

$$\text{rank}(S^{(i)}) \leq 2^i \tag{S25}$$

Lemma 2. If a matrix $S^{(i)}$ is constructed from s using Eq. (S3), and we define an $2^i \times 2^i$ square matrix $\hat{S}^{(i)}$ by:

$$\hat{S}^{(i)}_{k,j} = S^{(i)}_{k,j} \quad k = 1, \ldots, 2^i \quad j = 1, \ldots, 2^i \tag{S26}$$

Then $\hat{S}^{(i)}$ has full rank.

Proof. Consider the equation system:

$$\sum_{k=1}^{2^i} c_k s_{i,j-1+k} = 0 \qquad \forall j = 1, \ldots, 2^i \tag{S27}$$

$\hat{S}^{(i)}$ has full rank, if and only if Eq. (S27) implies that all c_k are 0, which is what we are going to show here. We can write Eq. (S27) for even and odd j, following Eq. (7) in the main text:

$$0 = \sum_{l=1}^{2^{i-1}} \left[c_{2l-1}(-1)^{\lceil \frac{i+2l-2}{2^i} \rceil} + c_{2l}(-1)^{1+\lceil \frac{i+2l}{2^i} \rceil} \right], \quad (j = 2p) \tag{S28}$$

$$0 = \sum_{l=1}^{2^{i-1}} \left[c_{2l-1}(-1)^{1+\lceil \frac{i+2l-1}{2^i} \rceil} + c_{2l}(-1)^{\lceil \frac{i-1+2l}{2^i} \rceil} \right], \quad (j = 2q-1) \tag{S29}$$

$$p = 1, \ldots, 2^{i-1} \tag{S30}$$

$$q = 1, \ldots, 2^{i-1} \tag{S31}$$

The right hand sides of Eqs. (S28) and (S29) are zero and thus equal to each other, yield the equation set (S32):

$$\begin{aligned} 0 = &\sum_{l=1}^{2^{i-1}} c_{2l-1} \left[(-1)^{\lceil \frac{p+l-1}{2^{i-1}} \rceil} + (-1)^{\lceil \frac{q+l-1}{2^{i-1}} \rceil} \right] \\ &- \sum_{l=1}^{2^{i-1}} c_{2l} \left[(-1)^{\lceil \frac{p+l}{2^{i-1}} \rceil} + (-1)^{\lceil \frac{q+l-1}{2^{i-1}} \rceil} \right] \end{aligned} \tag{S32}$$

We can determine the c_k from Eq. (S32) by iteratively looking at specific values of p and q. We perform the following iteration:

Initial conditions for iteration

Consider $p = 2^{i-1}, q = 2^{i-1}$, then Eq. (S32) is:

$$0 = -c_1 + \sum_{l=2}^{2^{i-1}} [c_{2l} - c_{2l-1}] \tag{S33}$$

Consider $p = 1, q = 2^{i-1}$, then Eq. (S32) is:

$$0 = c_1 - c_2 + 2_{2^i} \tag{S34}$$

Consider $p = 2^{i-1}, q = 1$, then Eq. (S32) is:

$$0 = c_1 \tag{S35}$$

Consider $p = 1, q = 1$, then Eq. (S32) is:

$$0 = \sum_{l=1}^{2^{i-1}-1} [c_{2l-1} - c_{2l}] + c_{2^{i-1}-1} \tag{S36}$$

Equations (S33), (S34), (S35), and (S36) together imply:

$$c1 = c2 = c_{2^i} = 0 \tag{S37}$$

With this knowledge, consider again $p = 1, q = 1$, then Eq. (S32) is:

$$0 = \sum_{l=2}^{2^{i-1}-1} c_{2l} - \sum_{l=2}^{2^{i-1}} c_{2l-1} \tag{S38}$$

Consider $p = 1, q = 2$, then Eq. (S32) is:

$$0 = \sum_{l=2}^{2^{i-1}-1} [c_{2l} - c_{2l-1}] + c_{2^i-1} \tag{S39}$$

Equations (S38) and (S39) imply:

$$c_{2^i-1} = 0 \tag{S40}$$

Iteration Step m.a
We know that:

$$c1 = 0, c_2 = 0, c_k = 0 \qquad \forall k > 2^i - 2(m-1) \tag{S41}$$

Consider $p = m - 1, q = m - 1$, then Eq. (S32) is:

$$0 = \sum_{l=2}^{2^{i-1}-m+1} [c_{2l} - c_{2l-1}] \tag{S42}$$

Consider $p = m, q = m - 1$, then Eq. (S32) is:

$$0 = \sum_{l=2}^{2^{i-1}-m} c_{2l} - \sum_{l=2}^{2^{i-1}-m+1} c_{2l-2m+1} \tag{S43}$$

Equations (S42) and (S43) imply:

$$c_{2^i-2m+2} = 0 \tag{S44}$$

Iteration Step m.b
Consider $p = m, q = m$, then Eq. (S32) is:

$$0 = \sum_{l=2}^{2^{i-1}-m} c_{2l} - \sum_{l=2}^{2^{i-1}-m+1} c_{2l-1} \tag{S45}$$

Consider $p = m, q = m + 1$, then Eq. (S32) is:

$$0 = \sum_{l=2}^{2^{i-1}-m} [c_{2l} - c_{2l-1}] + c_{2^i-2m+1} \tag{S46}$$

Equations (S45) and (S46) imply:

$$c_{2^i-2m+1} = 0 \tag{S47}$$

(Iteration finished)
By repeating steps m.a and m.b iteratively for $m = 2, \ldots, 2^{i-1} - 1$ it is shown that:

$$c_k = 0 \qquad \forall k = 1, \ldots, 2^i \tag{S48}$$

And thus, $\hat{S}^{(i)}$ must have full rank.
Remark: It has been observed empirically that the determinant of $\hat{S}^{(i)}$ is $\det(\hat{S}^{(i)}) = 2^{2^i-1}$ for $i = 1, \ldots, 8$, but this relationship has not been proven in general. □

Note that $S^{(i)}$ has at least the same rank or higher rank than $\hat{S}^{(i)}$ in Lemma 2 and therefore:

$$\text{rank}(S^{(i)}) \geq \text{rank}(\hat{S}^{(i)}) = 2^i \tag{S49}$$

Equations (S25) and (S49) prove Eq. (S4).

C. Phase invariant orthogonality and the Kronecker product

In this section we prove that if s is phase invariant orthogonal, then $t = s \otimes v$ is also phase invariant orthogonal for any arbitrary row vector v. Let m be the length of v and $k = pm + q$, $p = 0, \ldots, n-1$, $q = 0, \ldots, m-1$ then we calculate:

$$\begin{aligned}
&\sum_{j=1}^{nm} t_{i,j} t_{i',1+(j+k)\%(nm)} \\
=&\sum_{j=1}^{n}\sum_{l=1}^{m} t_{i,m(j-1)+l} t_{i',1+(m(j-1)+l+k)\%(nm)} \\
=&\sum_{l=1}^{m} v_l v_{1+(l+q)\%m} \sum_{j=1}^{n} s_{i,j} s_{i',1+(j+p)\%n} \\
=&0 \qquad \forall i \neq i'
\end{aligned} \tag{S50}$$

Eq. (S51) has been used in Eq. (7) in the main text.

D. Phase invariant orthogonality and the Kronecker product

In this section we prove that if s is phase invariant orthogonal, then $t = s \otimes v$ is also phase invariant orthogonal for any arbitrary row vector v. Let m be the length of v and $k = pm + q$, $p = 0, \ldots, n-1$, $q = 0, \ldots, m-1$ then we calculate:

$$\begin{aligned}
&\sum_{j=1}^{nm} t_{i,j} t_{i',1+(j+k)\%(nm)} \\
=&\sum_{j=1}^{n}\sum_{l=1}^{m} t_{i,m(j-1)+l} t_{i',1+(m(j-1)+l+k)\%(nm)} \\
=&\sum_{l=1}^{m} v_l v_{1+(l+q)\%m} \sum_{j=1}^{n} s_{i,j} s_{i',1+(j+p)\%n} \\
=&0 \qquad \forall i \neq i'
\end{aligned} \tag{S51}$$

Eq. (S51) has been used in Eq. (7) in the main text.

3. FURTHER DETAILS ON THE SURFACE NORMAL INTEGRATION

The Fast Marching method relies on updating the smallest value of the gradient within the grid first and then on updating the next smallest value. As it is not certain that the gradient of the depth Z only has one global minimum, the Eikonal equation is introduced for a function W of the form [1]:

$$W = Z + \lambda f \tag{S52}$$

where Z is the height, λ a constant and f a squared-distance function that is determined as $f = (x - u_x)^2 + (y - u_y)^2$ with (u_x, u_y) the starting point coordinates, which are set manually. The purpose of the function f is to deal with the global minimum value issue of any gradient grid. In fact, f will cancel all the critical points of the grid by creating one global minimum value: (u_x, u_y). By introducing P and Q as : $P = \nabla Z_x = N_x/N_z$ and $Q = \nabla Z_y = N_y/N_z$ which represent the gradient of the surface Z of the object, the Eikonal equation can be determined by solving:

$$\nabla W_x = P + 2\lambda x \tag{S53}$$

$$\nabla W_y = Q + 2\lambda y \tag{S54}$$

where ∇W_x and ∇W_y are the gradient of W in x and y. Once W is retrieved then it is easy to recover Z from Eq. (S52). We implemented the algorithm in Matlab™ following [1–5]

A. Fast Marching methods: 3D reconstruction comparison with Yvain Queau data set

In order to assess our algorithm for the Fast Marching method, we used a data set made available by Yvain Queau *et al.* [6] which consist of the mask of the vase, the gradient values of a vase: P and Q where $P = N_x/N_z$ and $Q = N_y/N_z$, and the ground truth of the vase. We use the gradients as inputs with our Fast Marching algorithm and we plot the result, see Fig. S1. By plotting the ground truth, we can then assess if the reconstruction is accurate. The RMSE between our reconstruction and the ground truth is equal to 1.8 a.u, as we don't have details about the calibration of the object and we can only estimate the error in arbitrary units. The most important point here is that the shape is exactly the same, hence we can conclude that the algorithm is working correctly.

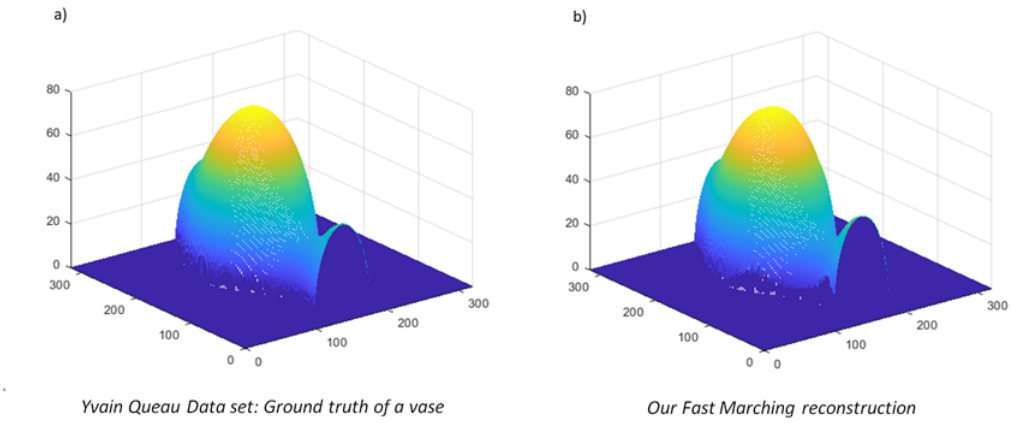

Fig. S1. Fast Marching Method. Comparison of a vase ground truth from a) Yvain Queau data set [6] with b) our 3D reconstruction using the Fast Marching method.

4. BENCHMARK WITH CONVENTIONAL *IN-PLANE CONFIGURATION*

For a spherical test object, Figs. S3 a)-d) show the four images obtained after decoding the frames in the conventional PS configuration that we call 'in-plane' (see Fig. S2). The reflectivity of each decoded image clearly demonstrates the different illumination direction left/bottom/right/top respectively for LED1, LED2, LED3 and LED4. According to the image scale, the amount of light for each direction is similar.

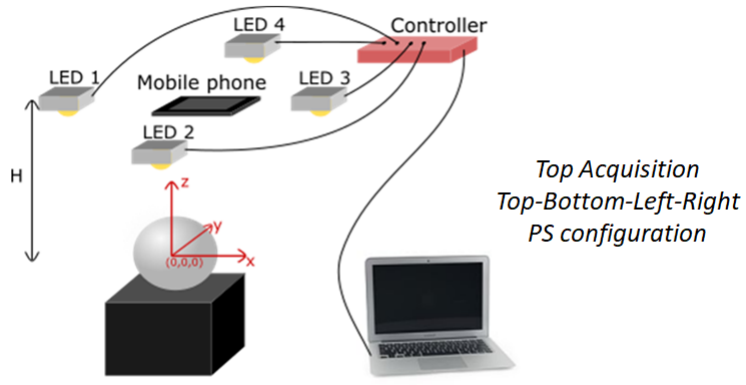

Fig. S2. Schematic of the photometric stereo imaging setup in the conventional configuration.

Figs. S3 e), f), g) show the corresponding surface normal components N_x, N_y, N_z. As expected from the 'in-plane' acquisition scheme in Fig. S2), N_x correctly distinguishes left and right facing surfaces of the object. Similarly, N_y correctly identifies up and down facing surfaces. Finally, as we cannot see the back of the object, N_z is always positive with some variations due to the depth

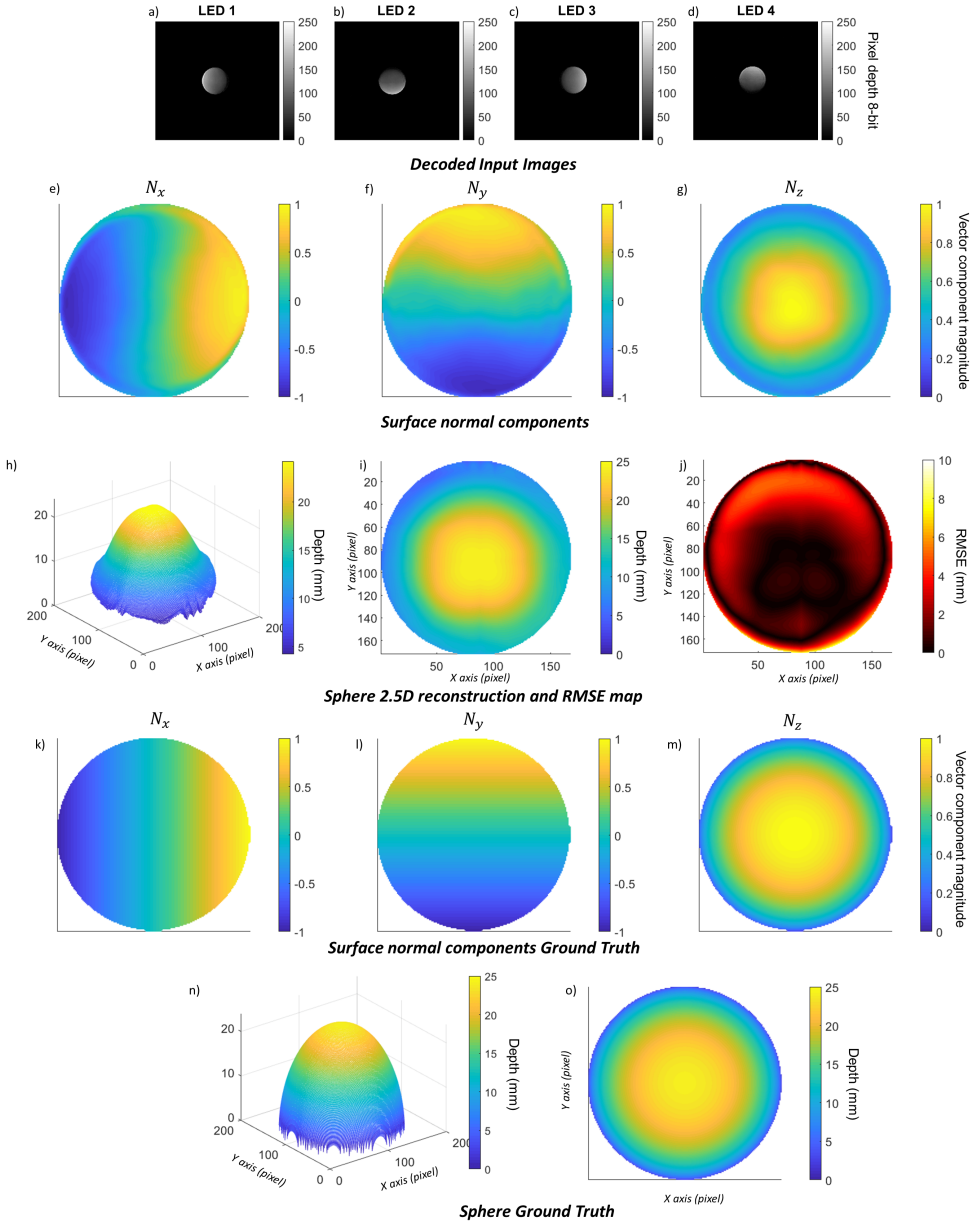

Fig. S3. 'In-plane' configuration results. a), b), c) and d) Decoded images after demodulation of the recorded frames, respectively for LED1, LED2, LED3 and LED4. e), f), g) Corresponding surface normal components N_x, N_y, N_z. h), i), j) 2.5D reconstruction of the sphere respectively perspective view, top view and RMSE error map. k), l), m), n) o) Sphere ground truth surface normal components and topography, respectively N_x, N_y, N_z, perspective view and top view.

of the object. By comparing it with the surface normal vectors of the ground truth of the sphere in Figs. S3 k), l), m), the match between N_x, N_y and N_z is satisfactory with some error on N_z. Indeed, we can see that N_z decreases slightly faster than the ground truth from its central maximum.

Two 2.5D reconstruction views and the RMSE error map are plotted in Fig. S3 h), i), j) respectively using Eq.(8) and Eq.(9) in maint text. For this 48 mm diameter sphere, we obtained an RMSE of 2.4 mm and an NRMSE of 5 %. These two results are within the same range of error as in [7]. Fig. S3 j) shows the RMSE map over the sphere in order to have a visualisation of the error distribution. We notice that the error is higher at the edge of the sphere compared to the middle area. This can be explained by the Fast Marching method as the error builds up as the reconstruction progresses through the surface. By comparing both the error map with the surface normal components from Figs. S3 e), f), g), we can see that the error on the 2.5D reconstruction is also closely linked to the error on N_z. When focusing on the top of the sphere, the reconstruction matches the ground truth from its central maximum down to 8 mm, which is deemed satisfactory.

REFERENCES

1. J. Ho, J. Lim, M.-H. Yang, and D. Kiriegma, "Integrating surface normal vectors using fast marching method," Comput. Vision-ECCV pp. 239–250 (2006).
2. J. A. Sethian, *Level Set Methods and Fast Marching Methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and material science*, Cambridge University (Cambridge monographs on applied and computational mathematics, 1999).
3. J. A. Sethian, "A fast marching level set method for monotonically advancing fronts." Proc. Natl. Acad. Sci. **93**, 1591–1595 (1996).
4. R. Sedgewick, *ALGORITHMS* (Addison-Wesley Publishing Company, 1983).
5. E. Ziegel, W. Press, B. Flannery, S. Teukolsky, and W. Vetterling, *Numerical Recipes: The Art of Scientific Computing* (Cambridge University, 1987).
6. Y. Quéau, J. D. Durou, and J. F. Aujol, "Normal Integration: A Survey," J. Math. Imaging Vis. **60**, 576–593 (2018).
7. Y. Zhang, G. M. Gibson, R. Hay, R. W. Bowman, M. J. Padgett, and M. P. Edgar, "A fast 3D reconstruction system with a low-cost camera accessory," Sci. Reports **5**, 1–7 (2015).