Inertial Safety from Structured Light

Sizhuo Ma and Mohit Gupta

University of Wisconsin-Madison, Madison WI 53706, USA
{sizhuoma,mohitg}@cs.wisc.edu

Abstract. We present inertial safety maps (ISM), a novel scene representation designed for fast detection of obstacles in scenarios involving camera or scene motion, such as robot navigation and human-robot interaction. ISM is a motion-centric representation that encodes both scene geometry and motion; different camera motion results in different ISMs for the same scene. We show that ISM can be estimated with a two-camera stereo setup without explicitly recovering scene depths, by measuring differential changes in disparity over time. We develop an active, single-shot structured light-based approach for robustly measuring ISM in challenging scenarios with textureless objects and complex geometries. The proposed approach is computationally light-weight, and can detect intricate obstacles (e.g., thin wire fences) by processing high-resolution images at high-speeds with limited computational resources. ISM can be readily integrated with depth and range maps as a complementary scene representation, potentially enabling high-speed navigation and robotic manipulation in extreme environments, with minimal device complexity.

1 Introduction

Imagine a drone flying through a forest or a robot arm repairing a complex machine part. In order to determine if they are on a collision course with an obstacle, they require knowledge of the 3D structure of the surroundings, as well as their own motion. Although classical approaches such as SLAM [5, 29] can recover 3D geometry and motion, doing so at high-speeds needed for fast collision avoidance is often prohibitively expensive. While a full 3D map and precise motion may be needed for long-term navigation policies (and for other applications such as augmented reality), it may not be critical for making short-term but time-critical decisions like detection and avoidance of obstacles.

In this paper, we propose *weak 3D cameras* which recover scene representations that are less informative than 3D maps, but can be captured considerably faster, with limited power and computation budgets. These weak 3D cameras are based on inertial safety maps (ISM), a novel scene representation tailored for time-critical and resource-constrained applications such as fast collision avoidance. ISM, for each pixel, is defined as the product of scene depth and time-to-contact (TTC) – time it will take the camera to collide with the scene if it keeps moving with the current velocity [19]. ISM is a *motion-centric scene representation*; it encodes *both* the 3D scene geometry as well as scene-camera relative motion.[1] Given a scene, different motions of the camera lead to different ISMs,

[1] In contrast, a 3D map is a motion-invariant scene representation.

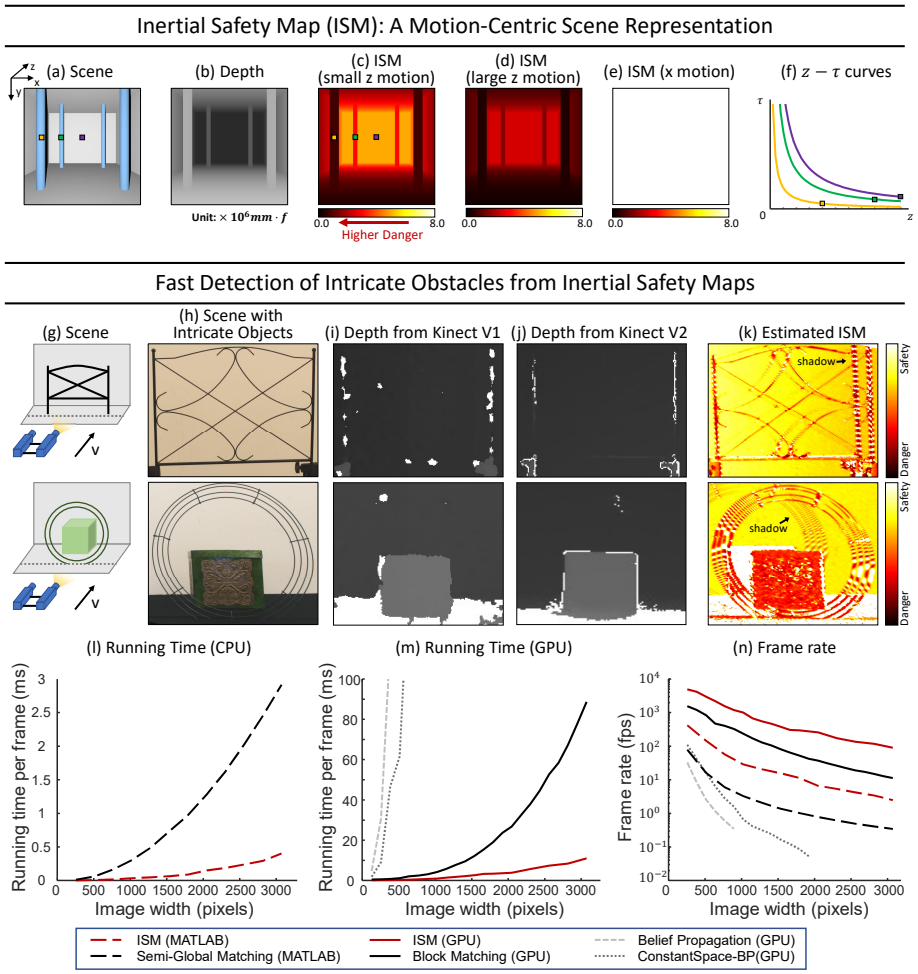

Fig. 1. Inertial Safety Map (ISM) is a motion-centric scene representation tailored for fast collision avoidance. **(a-b)** An example scene – a room with several pillars. **(c-e)** For the same scene, different camera motion results in different ISMs. A low value of ISM indicates a higher likelihood of collision, whereas higher values convey safety in the immediate future. **(f)** For a given value of ISM, the possible (z, τ) pairs lie on a hyperbolic curve called the $z-\tau$ curve, which can be used for navigation policy design. **(g-h)** Scenes with intricate objects that are a few millimeters thick being observed from a distance of 1.5m. **(i-j)** Conventional depth cameras based on structured-light or time-of-flight have low spatial resolution, and cannot resolve these thin objects. **(k)** The proposed ISM estimation algorithm, due to its low computational complexity, can use high-resolution images for detecting intricate obstacles, while maintaining high speeds. With conventional 3D imaging techniques, increasing the resolution comes with increased device complexity or high computational cost, often precluding real-time performance. **(l-n)** Quantitative timing comparisons show that for the same image resolution, the unoptimized CPU, and the GPU implementation of the proposed method are up to one order of magnitude faster than existing matching methods.

as shown in Fig. 1(a-e). ISM lends itself to intuitive interpretations that can be readily incorporated in robot navigation policies; small ISM indicates potentially imminent danger of collision, whereas large values convey relative safety.

Active ISM using structured illumination: Consider a robot equipped with a stereo camera pair. Our key insight is that it is possible to directly recover the ISM without explicitly computing the disparities and depths, by performing a differential analysis of stereo image formation. Based on this, we develop a theoretical model of active ISM, the structured light (SL) counterpart of stereo-based ISM, where one of the cameras from the stereo pair is replaced with a projector (which is treated as an inverse camera). The projection of coded intensity patterns enables robust estimation of ISM even in challenging scenarios, such as textureless and geometrically complex objects. Based on the active ISM model, we develop a practical *single-shot* ISM recovery method that requires projecting and capturing only a single image[2]. This can be readily implemented with low complexity optical devices (e.g., an LED with a static mask), and is amenable to high-speed motion scenarios.

Fast detection of intricate obstacles: Single-shot structured light methods typically require computationally expensive algorithms for computing correspondences. Since ISM requires only *differential disparity* and not absolute correspondences, we design a fast algorithm based on Fourier analysis of the images, which enables real-time estimation of ISM even for high-resolution images using only commodity hardware with a limited computational budget. As a result, the proposed approaches can detect intricate obstacles (e.g., tree twigs, thin wire fences) that are beyond the capabilities of commodity 3D cameras [43], which have low resolution due to constraints on device complexity and computational resources (Fig. 1(g-k)). While it is theoretically possible to increase the resolution of conventional 3D imaging techniques, it often comes with a high computational cost. For example, we perform timing comparisons of CPU-based MATLAB and GPU-based CUDA C++ implementations of our approach, both of which are up to one order of magnitude faster than current methods *at the same spatial resolution* (Fig. 1(l-n)). With such computational benefits across computing architectures, ISMs can help robots with limited computation budgets navigate challenging environments with intricate obstacles.

Scope and limitations: ISM should not be seen as a replacement for conventional scene representations such as depth maps, which are needed for long-term path planning. Instead, ISM should be considered a complementary representation that can be recovered at lower time and power budgets, but only provides conservative collision estimation. In general, it is not possible to recover depth maps from ISMs. However, it is possible to recover depth map from the same captured data used for estimating single-shot active ISM, albeit with a higher computational cost. In future, we envision navigation policies with parallel threads utilizing the same data – a *fast* thread to estimate ISM that makes fast navigation decisions such as braking and collision avoidance, and a *slow* thread to create a full 3D map to aid high-level navigation. Developing such policies, although beyond the scope of this paper, is an important next step.

[2] The method is "single-shot" in that we compute N ISMs from $N+1$ frames (single-shot except one initial frame).

2 Related Work

Single-shot structured light 3D imaging. Single-shot structured light (SL) methods project only one pattern to recover depths and thus are suitable for dynamic scenes. Common patterns include sinusoids [36, 37], de Bruijn stripes [23, 33], grids [21, 34], or random dots [43]. These methods often rely on computationally expensive search algorithms, and cannot operate at high frame rates. Recently, Fanello *et al.* [6] treated SL matching as a classification problem and showed depth recovery at 1kHz for 1MP images. Our goal is different: We define a motion-centric safety measure that is fundamentally easier to compute than depths. An interesting next step is to apply learning techniques to further increase the efficiency of ISM computation. Furukawa *et al.* [9] has a similar idea of utilizing the disparity change due to object motion, but require one or more color projectors, which increases the hardware complexity and reduces robustness for scenes with non-uniform color distributions [38].

Collision detection based on other modalities. Proximity sensors based on various modalities (LiDAR [12], ultrasound [35], RADAR [1], programmable light curtain [2, 40]) either measure a single global proximity value or require mechanical scanning for generating a 2D map. Time-of-flight (ToF) [15] and Doppler ToF-based methods [16] require correlation sensors. Because of their hardware complexities, all these methods are usually limited in resolution (see Fig. 1 for examples). Navigation based on optical flow [4, 11] and time-to-contact [19, 28, 41] uses passive sensors and is not suitable for textureless scenes and low-light environment. Our method is active, single-shot, has low hardware and computational complexity, and recovers a high-resolution 2D safety map with fine details that can be used for avoiding obstacles with thin structures.

Metrics used for collision avoidance. The level of safety for a robot to navigate without collision depends on not only distance to obstacles (depth map), but also speed, mass, physical size, *etc* [27]. In robotics, several works [17, 20, 22, 44] have proposed safety metrics for collision avoidance. In this paper, we propose a novel safety metric that captures both the geometry and the motion aspects of safety, and can be estimated from visual data with minimal computation requirements. Deploying this metric in real-world robotic applications is an exciting direction for future work.

3 Inertial Safety Map

In this section, we present *inertial safety map* (ISM), a novel representation of the scene for collision avoidance in scenarios involving scene or camera motion (e.g., robot navigation). Consider a rectified binocular stereo setup observing a scene. Suppose a scene point $\mathbf{R} = (x, y, z)$ projects to pixel (u, v) in the right view, and pixel $(u + \Upsilon, v)$ in the left view. Υ is called the disparity of $\mathbf{R}$. The disparity Υ and the depth z of $\mathbf{R}$ are related by the triangulation equation:

$$z = \frac{fb}{\Upsilon}, \tag{1}$$

where f is the focal length of the cameras (assumed same for both cameras). b is the baseline of the stereo setup. Stereo algorithms compute depths z by estimat-

ing corresponding pixels and disparity Υ between the stereo image pair, which often requires computationally intensive search and optimization algorithms [6].

3.1 Differential Analysis of Triangulation Equation

Suppose the scene point $\mathbf{R}$ moves with respect to the camera pair due to scene/camera motion. Due to this relative motion, the disparity Υ may change over time. Our key observation is that, although computing absolute disparities Υ may be expensive, it is possible to efficiently recover *differential changes in disparity* $\Delta\Upsilon$ due to small motion. For instance, a differential disparity change may be estimated by searching in a small local window instead of the entire epipolar line [3]. Later in Section 5, we will discuss an approach for fast computation of differential disparity change in an active stereo (coded structured light) system.

What information is recoverable from disparity change? We address this question by performing a differential analysis of the triangulation equation (Eq. 1). By re-writing $\Upsilon = \frac{fb}{z}$ as a function of depth z from Eq. 1, and taking the derivative of Υ with respect to time t, we get:

$$\frac{\mathrm{d}\Upsilon}{\mathrm{d}t} = -\frac{fb}{z^2}\frac{\mathrm{d}z}{\mathrm{d}t}\,. \tag{2}$$

Assuming the time difference Δt between two successive frames is small, we can multiply both sides by Δt and get

$$\Delta\Upsilon = -\frac{fb}{z^2}\Delta z\,, \tag{3}$$

where $\Delta\Upsilon$ and Δz are the changes in the disparity and depth of point $\mathbf{R}$, respectively, due to relative scene-camera motion. Next, we define *time-to-contact* $\tau = -\frac{z}{\Delta z}$, as the time it will take for the camera (the image plane of the right camera) to collide with point $\mathbf{R}$ if the relative velocity between the camera and the point remains the same [19]. By substituting in the above equation, and rearranging the terms, we get the following key relationship:

$$\boxed{z \cdot \tau = \frac{fb}{\Delta\Upsilon}\,.} \tag{4}$$

Assuming a calibrated stereo system (known focal length f and baseline b), the right-hand side involves only one unknown $\Delta\Upsilon$, which we assume can be computed efficiently. The left-hand side is the product of two quantities that are indicators of the chances of keeping moving safely. Intuitively, a large value of the product of z and τ indicates low chances of collision. Based on this intuition, we define the *inertial safety measure* for collision avoidance as follows:

Definition. The *inertial safety measure* S of a scene point with respect to its relative motion to the stereo camera is defined as the product of its depth z and time to contact τ,

$$S = z \cdot \tau = \frac{fb}{\Delta\Upsilon}\,. \tag{5}$$

which can be computed from camera parameters f and b, and disparity change $\Delta\Upsilon$. The *inertial safety map (ISM)* is a per-pixel map of inertial safety measure.

The inertial safety measure S encodes the level of safety *if the camera keeps its current motion*. By estimating $\Delta\mathcal{Y}$, we can compute S for collision avoidance. For example, a robot can detect obstacles and get around them by identifying image regions with low values of S, without explicitly computing depths.

3.2 Inertial Safety Map: Interpretations

Imagine a fast-moving drone navigating around pillars in a room (Fig. 1). For this simulated scene, we plot the ground truth ISMs for three different motions. The unit of ISM is $mm \times f$, where depth is in mm and the time-to-contact (TTC) is expressed in terms of the number of frames before collision. Darker colors represent low values of the ISM, and therefore a higher level of danger of collision. All values higher than a threshold, or less than zero (due to camera moving away from the scene) are mapped to white.

A motion-centric scene representation: From Figs. 1(c-e), we observe that for the same scene, different motions results in different ISMs. This is because the TTC depends on the z-velocity. The amount of z-motion between frames is doubled in (d) compared to (c), so the TTC is halved for every pixel. In (e), since there is no z-motion, the ISM is $+\infty$ everywhere. Thus, the inertial safety map can be considered a *motion-centric scene representation* as it depends both on the scene's geometry, as well as the relative scene-camera motion; it encodes the degree of safety (from collision) if the camera/scene keeps its *current motion*.

$z-\tau$ **curve:** A given value of the ISM corresponds to an infinite number of possible $z-\tau$ pairs, which trace out a hyperbolic curve called the $z-\tau$ curve in the 2D $z-\tau$ space (Fig. 1(c,f)). The $z-\tau$ curves corresponding to three highlighted scene points are plotted in (f), with the exact (z,τ) values indicated by the colored rectangles. Although we cannot determine the true z and τ values from an estimate of the ISM, the ISM can be used as a fast and conservative safety check in robot navigation policies. This is because when ISM is high, both z and τ have to be high, so the robot is safe. When ISM is low, it can be due to either high z and low τ, or low z and high τ. This ambiguity can be resolved by designing a more sophisticated navigation policy, or by triggering a full depth recovery algorithm for collision avoidance. See the supplementary technical report for a detailed discussion.

4 Active Inertial Safety Map

So far we have defined the inertial safety map for a passive two-camera stereo system. However, the ISM can be generalized to any two-view imaging system, including active methods such as structured light (SL), where the second camera is replaced by a projector (an inverse camera). In SL, a coded light pattern is projected to enable robust scene recovery even in challenging scenarios including lack of scene texture and insufficient lighting. In this section, we develop mathematical model and approaches for *active ISM*, i.e., recovering ISM using SL. Specifically, we consider active ISM recovery from *single-shot SL* where a single image is captured with a single projected pattern.

Consider a projector-camera system with a horizontal baseline For ease of analysis, we assume that the projector projects a pattern with 1D translational

symmetry, i.e., all the projector pixels in a column have the same intensity. Such patterns are used in several SL 3D imaging systems [23, 37], and can be expressed as a 1D function $P(c)$, where c is the projector column index.

Suppose a scene point $\mathbf{R}$ is illuminated by projector column index c, and imaged at camera pixel (u, v) at time t. The intensity of pixel (u, v) at t is:

$$i(u, v, t) = \alpha(u, v, t)\, P(c) + \beta(u, v, t)\,, \tag{6}$$

where $\alpha(u, v, t)$ encapsulates the reflectance properties of point $\mathbf{R}$, and $\beta(u, v, t)$ is the intensity component due to ambient light. The projector column index c, camera pixel index u and the disparity Υ are related as:

$$\Upsilon(u, v, t) = c - u. \tag{7}$$

Suppose point $\mathbf{R}$ moves with respect to the camera (due to camera or scene motion) from time t to $t + \Delta t$. After motion, let the point be illuminated by projector column index $c + \Delta c$, and imaged at camera pixel $(u + \Delta u, v + \Delta v)$. Similar to Eq. 6, the observed intensity of point $\mathbf{R}$ at $t + \Delta t$ is given as:

$$i(u + \Delta u, v + \Delta v, t + \Delta t) = \alpha'\, P(c + \Delta c) + \beta'\,, \tag{8}$$

where $\alpha' = \alpha(u + \Delta u, v + \Delta v, t + \Delta t), \beta' = \beta(u + \Delta u, v + \Delta v, t + \Delta t)$. The disparity of point $\mathbf{R}$ after motion is then:

$$\Upsilon(u + \Delta u, v + \Delta v, t + \Delta t) = (c + \Delta c) - (u + \Delta u)\,. \tag{9}$$

Recovering the ISM requires measuring the disparity change $\Delta\Upsilon$, which is the difference between the new and old disparity, i.e., $\Delta\Upsilon = \Upsilon(u+\Delta u, v+\Delta v, t+\Delta t) - \Upsilon(u, v, t)$. From Eqs. 7 and 9, we get:

$$\Delta\Upsilon = \Delta c - \Delta u\,. \tag{10}$$

Computational considerations: To compute $\Delta\Upsilon$, we need to estimate both the "texture flow" (Δu) and the "illumination flow" Δc, which are the projected motion of the scene point on the camera's and projector's image planes [39]. This problem is challenging due to several non-linearly coupled unknowns for each pixel $(\alpha, \beta, \Upsilon, \Delta u, \Delta c)$. Solutions require expensive nonlinear optimization and therefore are not suitable for applications with limited computational budget and extreme timing requirements.

To make the computation tractable, instead of considering the disparity change $\Delta\Upsilon$ of a *fixed scene point*, we consider $\Delta\Upsilon$ of a *fixed pixel* in the camera image. At time $t + \Delta t$, we analyze the image intensity at the *same* pixel (u, v):

$$i(u, v, t + \Delta t) = \alpha(u, v, t + \Delta t)\, P(c + \Delta\Upsilon_R) + \beta(u, v, t + \Delta t)\,,$$

where $\alpha(u, v, t + \Delta t)$ and $\beta(u, v, t + \Delta t)$ are the reflectance and ambient terms for the scene point imaged at pixel (u, v) after motion, and

$$\Delta\Upsilon_R = \Upsilon(u, v, t + \Delta t) - \Upsilon(u, v, t) \tag{11}$$

is the disparity change along the camera ray at pixel (u, v). This definition of *active ISM* does not compute correspondences between frames, instead relying

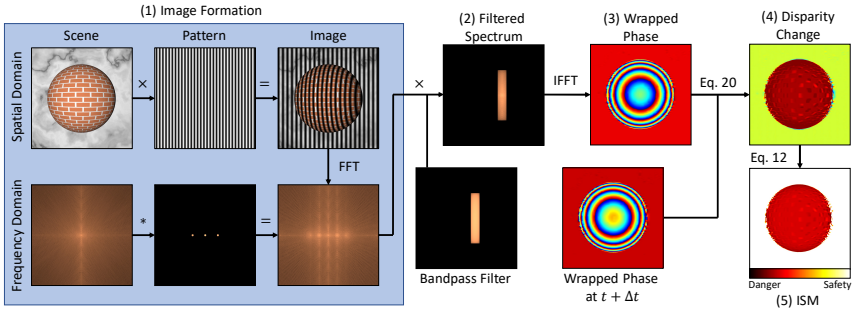

Fig. 2. Overview of active ISM recovery method. (1) The scene is illuminated by a high-frequency sinusoidal pattern, which can mathematically be represented as a multiplication in the spatial domain and a convolution in the frequency domain (ignoring ambient light). The frequency domain images are plotted in log scale. (2) A bandpass filter is applied to extract the term G. (3) Inverse FFT is used to get wrapped phase maps at time t and $t + \Delta t$. (4) Combine both maps to get the disparity change (Eq. 19). (5) Compute ISM (Eq. 5).

on a differential analysis that estimates the differential depth change between two frames at each pixel. The resulting active ISM provides a *conservative* measure of danger that detects all potential collisions: When a collision is about to happen, the depth at the corresponding pixel will decrease to zero. Therefore, the ISM will be small at the pixel at some point before collision. As we show in Section 5, it is possible to estimate $\Delta\Upsilon_R$ with simple (linear) analytic expressions that can be computed extremely fast with limited computational resources.

ISM estimation under sharp depth variations: Due to relative scene-camera motion, pixel (u, v) may image different scene points at times t and $t + \Delta t$. As a result, the computed ISM value may result in overly conservative collision warnings, especially at depth edges where the depth changes significantly across frames. This issue can be mitigated by spatio-temporal filtering the estimated ISM. See the supplementary report for a detailed discussion.

5 ISM from Single-Shot Structured Light

In this section, we present practical approaches for computing active ISM from single-shot structured light. One way to estimate ISM is to directly estimate scene disparities by using globally-unique patterns such as de Bruijn [23, 33] and random patterns [6]. Once disparities Υ are computed before and after motion, ISM can be trivially computed by taking their difference (Eq. 11). However, single-shot SL methods typically requires computationally intensive algorithms which are not suitable for scenarios with limited computational budget.

5.1 Fast Fourier Domain Computation of ISM

We propose a fast method for computing ISM, based on projecting a 1D high-frequency sinusoid pattern. A pictorial summary of the method is shown in

Fig. 2. Let the projected sinusoid pattern be given as:

$$P(c) = 0.5 + 0.5\cos(\omega c), \tag{12}$$

where ω is the angular frequency of the sinusoid. Substituting in Eq. 6, the intensity at pixel (u, v) is:

$$i = \alpha + \alpha\cos(\omega u + \omega\Upsilon) + \beta\,, \tag{13}$$

where abusing the notation, the constant 0.5 is absorbed with α. For brevity, we drop the indices (u, v, t). Eq. 13 is an underconstrained nonlinear equation in three unknowns (α, β and Υ), and thus, challenging to solve directly.

Solving Eq. 13 by linearizing and regularizing: The cos term on the right hand side can be expanded into a sum of complex functions:

$$\begin{aligned} i &= \alpha + \alpha\cdot(e^{j\omega(u+\Upsilon)} + e^{-j\omega(u+\Upsilon)})/2 + \beta \\ &= \underbrace{\alpha+\beta}_{f} + \underbrace{0.5\,\alpha\,e^{j\omega\Upsilon}}_{g} e^{j\omega u} + \underbrace{0.5\,\alpha\,e^{-j\omega\Upsilon}}_{g^*} e^{-j\omega u}, \end{aligned} \tag{14}$$

where $j = \sqrt{-1}$. The above is now a linear equation in three unknowns f, g and g^* (conjugate of g):

$$i(u, v, t) = f(u, v, t) + e^{j\omega u}\, g(u, v, t) + e^{-j\omega u}\, g^*(u, v, t). \tag{15}$$

Regularizing by assuming global smoothness: One way to solve Eq. 15 is to make the restrictive assumption that the variables f, g and g^* are locally constant and stack the equations in a local neighborhood into a linear system (similar to Lucas-Kanade image alignment [26]). However, inspired by Fourier transform profilometry [36, 37], we make a less restrictive assumption that scene reflectance and depths (and thus, α, β and Υ) vary smoothly horizontally (in each row) compared to the pattern frequency ω. In this case, f and g are band-limited signals. Taking the 2D Fourier Transform with respect to u and v:

$$\begin{aligned} I(\omega_u, \omega_v, t) =& F(\omega_u, \omega_v, t) + G(\omega_u - \omega, \omega_v, t) \\ &+ G^*(\omega_u + \omega, \omega_v, t)\,. \end{aligned} \tag{16}$$

The spectra F, G and G^* can be separated from each other by ω, as shown in Fig. 2. We extract the spectrum $G(\omega_u - \omega, \omega_v, t)$ by applying a bandpass filter (a 2D Hanning window as in [24]) and transform it back to the primal domain. The complex signal g is recovered as:

$$g(u, v, t) = 0.5\,\alpha(u, v, t)\, e^{j\omega\Upsilon(u,v,t)}. \tag{17}$$

The disparity can be estimated from the complex argument:

$$\hat{\Upsilon} = \arg(g(u, v, t))/\omega\,. \tag{18}$$

Oriented 2D filter: Consider a tilted thin thread in front of a wall, as shown in Fig. 3 (a). The disparity no longer changes smoothly in each row, which makes the spectra inseparable by using simple vertical bandpass filters (Fig. 3 (b)).

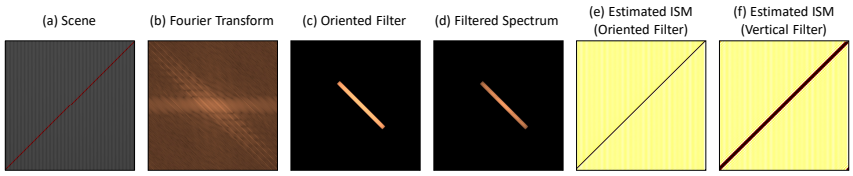

Fig. 3. Oriented 2D filter for resolving thin structures with different orientations. (a) A tilted thin thread in front of a wall. Disparity changes abruptly in each row. **(b-d)** The spectra are no longer separable by using simple vertical bandpass filters. Instead, they can be separated using oriented filters. **(e-f)** An oriented filter recovers the ISM with higher accuracy as compared to a vertical filter.

However, if we filter the spectrum using an oriented 2D filter, as shown in (c), it is still possible to separate the signal G from F, as shown in Fig. 3 (d). As a result, the estimated ISM (Fig. 3 (e)) is more accurate than using a vertical filter (Fig. 3 (f)). In practice, it is possible to divide the image into small patches, and filter each patch using different oriented filters to detect thin structures with different orientations. See the supplementary report for details.

Need for phase unwrapping? It may appear from first glance that absolute disparity Υ can be recovered from Eq. 18. To recover the absolute disparity, one needs to recover the absolute phase $\phi_u = \omega\Upsilon$. However, we can only recover the wrapped phase ϕ_w, related to ϕ_u by $\phi_u = \phi_w + 2k\pi$ for some unknown $k \in \mathbb{N}$.[3] Absolute phase ϕ_u can be recovered by spatial phase unwrapping methods [10], which require global reasoning and are highly computationally intensive.

Fortunately, to compute the ISM S_E, we only need to compute the disparity change $\Delta\Upsilon_R$, instead of the absolute disparity Υ. Assuming a small change in Υ across consecutive frames, *i.e.*, $\Delta\Upsilon_R \in (-\pi/\omega, \pi/\omega]$, it is possible to compute $\Delta\Upsilon_R$ by taking the difference of wrapped phases:

$$\Delta\Upsilon_R(u,v,t) = {}^{\text{wrap}(\arg(g(u,v,t+\Delta t))-\arg(g(u,v,t))}/_{\omega}\,, \tag{19}$$

where $\text{wrap}(\phi)$ is a function that wraps a phase to the principal values. As a result, a dense S_E map can be computed efficiently *without phase unwrapping.*

5.2 Practical Considerations

Computational efficiency: ISMs can be computed at high speeds even for very high resolution images, as demonstrated in Fig. 1(l-n). A direct comparison between ISM and other single-shot SL methods is difficult since their code is usually not publicly available. Instead, we compare the computational speeds of the proposed ISM algorithm and a few widely-used stereo matching algorithms. The CPU (MATLAB) implementation of the method is up to one order of magnitude faster than MATLAB's semi-global matching algorithm [18]. We also develop a GPU implementation of the proposed method, which is able to reach 1kfps at 1 megapixel resolution, and achieves real-time performance even for very high

[3] It is possible to recover absolute phase using a unit-frequency sinusoid, however at a considerably lower phase-recovery precision than high-frequency sinusoids.

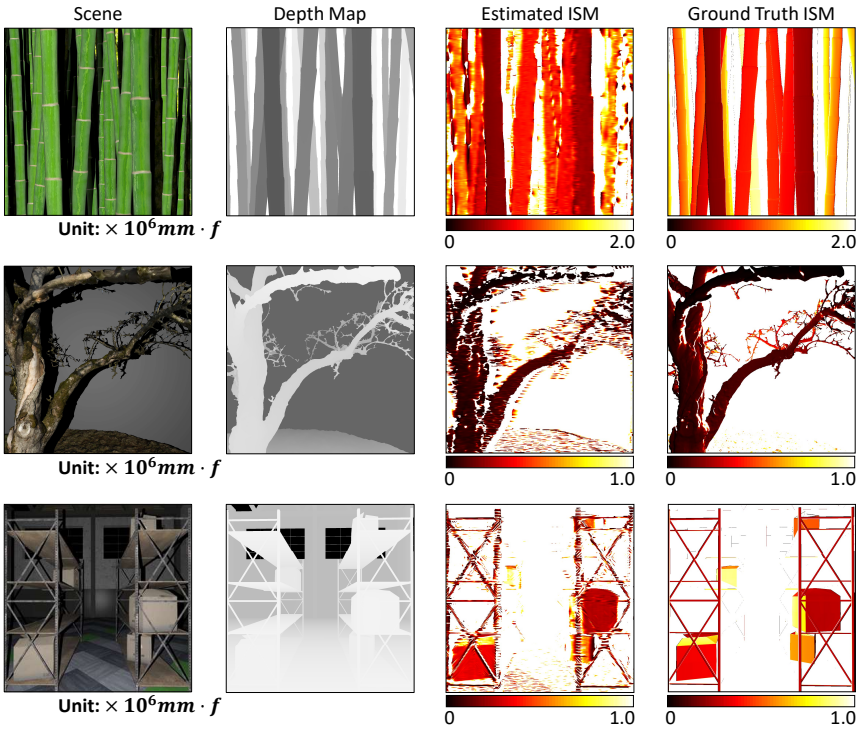

Fig. 4. Simulation results for example robot navigation scenarios through intricate obstacles. The proposed approach can recover the ISM for scenes with complex, overlapping thin structures (bamboos, tree branches, warehouse racks).

resolution (90fps at 9 megapixel), which is 9x faster than OpenCV's CUDA implementation of block matching and considerably faster than belief propagation (BP) [7] and constant-space BP [42]. See the supplementary report for details on the experiment setting. These comparisons demonstrate the computational benefit of ISM for both computational architectures. In practice, the exact implementation needs to be tailored to the available computing resources.

Allowable range of inter-frame motion vs. pattern frequency: The maximum inter-frame motion is determined by the recoverable disparity change $\Delta\Upsilon_R$, which is constrained by the pattern period $\Lambda = \frac{2\pi}{\omega}$ as $\Delta\Upsilon_R < \frac{\Lambda}{2}$. A low pattern frequency enables recovering a wider range of $\Delta\Upsilon_R$, thus allowing faster camera motion. Typical velocities for the current hardware prototype are 1-10 cm/s (captured at 30fps). On the other hand, using a higher frequency pattern can separate f and g with wider frequency bands, which means higher robustness for scenes with high-frequency textures or cluttered geometry. Finally, although our image model (Eq. 6) assumes only direct lighting, in practice, inter-reflections may result in erroneous estimates of S_E map. Using a high-frequency pattern also mitigates this effect [13, 30].

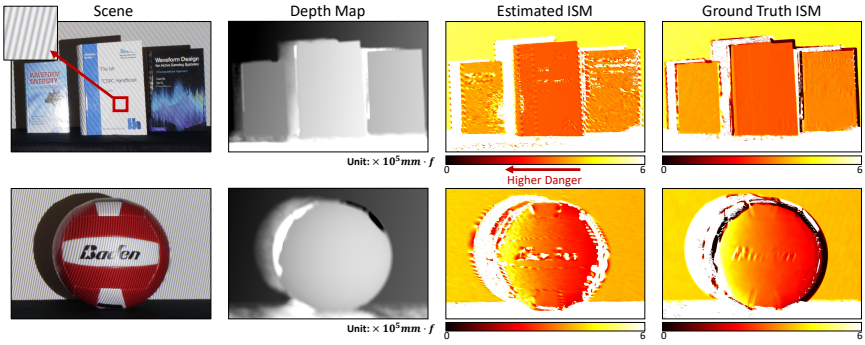

Fig. 5. Ground truth comparison. Our prototype structured light system consists of a Canon DSLR camera and an Epson 3LCD projector. The projector projects a 1920×1080 high-frequency sinusoidal pattern with a period of 8 pixels. **Zoom in to see the pattern.** ISMs estimated using the proposed method are compared with ground truth, which is obtained by projecting binary SL patterns using the same hardware. Depth maps of the scenes are also shown for comparison (not used in computing ISM). **(Top)** A piecewise planar scene consisting of three books. **(Bottom)** A spherical ball. Our method recovers the ISMs of both scenes accurately.

6 Experimental Results

Simulations. Fig. 4 shows three simulated scenes that emulate different robot navigation scenarios. Our method is able to estimate the ISM of the thin, complex geometry of bamboos, tree branches and warehouse racks.

Experiment setup. We build a prototype structured light system using a Canon DSLR camera and an Epson 3LCD projector. The projector projects a 1920×1080 high-frequency sinusoidal pattern with a period of 8 pixels. After rectification, a 3714×2182 captured image is used for ISM computation. The camera-projector baseline is 353mm. Details on calibration and rectification can be found in the supplementary technical report.

Ground truth comparison. Fig. 5 shows the comparison between the ISM estimated by our method and the ground truth, which is obtained by projecting a sequence of binary-coded SL patterns. The camera translates for roughly 3mm along the z-axis. Our method correctly estimates the ISM for scenes with planar and curved surfaces with strong depth edges.

Resolving extremely thin structures. Fig. 1(g-k) demonstrates our method's capability of recovering thin structures by processing high-resolution images, which is possible due to the hardware simplicity of structured light and the computational efficiency of the proposed algorithm. The thinnest part of the scenes are 4mm and 1.5mm, which could be challenging to resolve from 1.5m away. We also show results for two commodity depth cameras: Kinect V1 and V2, whose spatial resolutions are 640×480 and 512×424 respectively. From the same distance, the depth cameras are only able to partially recover the thicker parts of the fence in the top scene and completely miss the rings in the bottom scene. This is not meant to be a direct comparison of the three approaches,

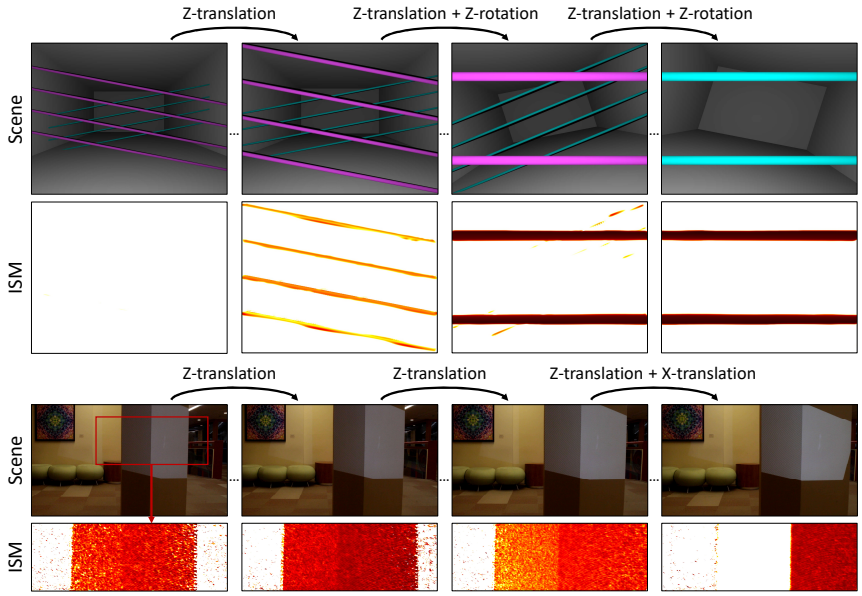

Fig. 6. Navigation sequences with manually planned trajectories. (Top) A simulated sequence where a drone flies through thin threads. As the drone detects the threads, it aligns its pose to be parallel with the threads to avoid collision. **(Bottom)** A real video sequence where a robot navigates around a pillar. The unrectified images are shown here to better convey the scene, while the ISM is only computed for the cropped area due to projector's field-of-view. The robot moves forward (first three frames), detects the pillar and moves to the left to circumvent it (last frame).

because the data is acquired from different cameras. With a higher resolution, depth cameras may also be able to recover the scene details, albeit at a higher computational cost, as shown in the timing comparisons in Fig. 1.

Navigation sequences. Fig. 6 shows a simulated sequence (top) where a drone flies through thin threads, and a real captured sequence which emulates a robot navigating around a pillar (bottom). Trajectories are manually planned in both examples. As the robot approaches the threads and the pillar, danger is detected from the estimated ISM, and the robot reacts accordingly to avoid collision.

Detecting object motion. ISMs can also be used to detect collision due to moving objects. For example, in a co-robot scenario where robots and human workers collaborate in the same environment [8, 25], it is important to prevent collisions between human and robot arms for safety. Fig. 7 shows two examples where a moving human hand and a thin cable are correctly detected in the ISMs. The thin cable is an intricate obstacle, which is challenging to detect with current depth cameras. The estimated ISMs can be used to avoid collision with human co-workers and intricate dynamic objects in the working environment. **All complete video sequences can be found in the supplementary video. See the technical report for additional results.**

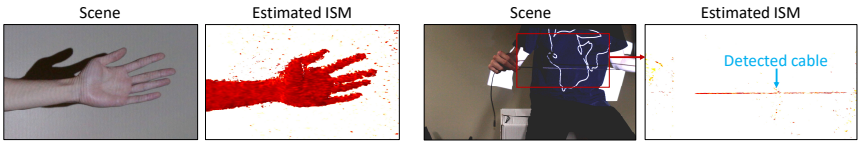

Fig. 7. Detecting object motion. ISMs can also be used to detect collisions between moving objects and a static camera. **(Left): A hand moving towards the camera. (Right): A thin cable (held by a person) moving towards the camera.** The unrectified images are shown here to better convey the scene, while the ISM is only computed for the cropped area due to projector's field-of-view.

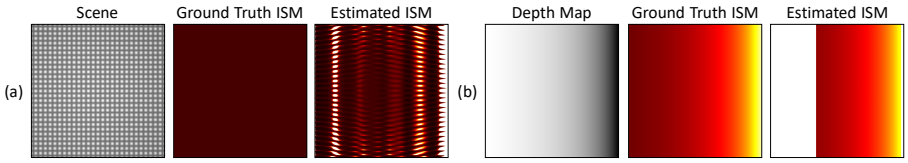

Fig. 8. Failure modes. Scene (a): High-frequency albedo. A plane with a very high-frequency bi-sinusoidal pattern, which violates the albedo smoothness assumption. **Scene (b): Fast motion at short range.** A slanted plane (see the depth map) moving fast in z-direction towards the camera. Disparity change in the closest part of the scene wraps around the period and the estimated ISM becomes negative (shown as white).

7 Limitations and Future Outlook

Failure modes. The proposed method may fail to estimate the ISM correctly when the assumptions made in Section 5.1 are not satisfied. This happens when the albedo varies too quickly, or when the objects are moving too fast at a short distance, causing disparity changes too abruptly (Fig. 8). See the supplementary technical report for a quantitative analysis.

Resolving vertical depth edges. Our method cannot accurately recover thin structures that are nearly vertical because the depth varies abruptly along the epipolar lines, which violates the smoothness assumption. A potential solution is to have one camera and two projectors in an L-configuration such that both horizontal and vertical epipolar lines are available.

Performance in outdoor settings: Outdoor deployment under sunlight is challenging for all power-limited active imaging systems due to photon noise from sunlight. It is possible to mitigate this issue by spatio-temporal illumination coding [14], as well as joint illumination and image coding [32, 31] to enable ISM recovery under strong sunlight.

Acknowledgement. This research is supported in part by the DARPA REVEAL program and a Wisconsin Alumni Research Foundation (WARF) Fall Competition award.

References

1. Azevedo, S., McEwan, T.E.: Micropower impulse radar. IEEE Potentials **16**(2), 15–20 (1997)
2. Bartels, J.R., Wang, J.: Agile Depth Sensing Using Triangulation Light Curtains. In: International Conference on Computer Vision (ICCV). pp. 7899–7907. IEEE (2019)
3. Čech, J., Sanchez-Riera, J., Horaud, R.: Scene flow estimation by growing correspondence seeds. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3129–3136. IEEE (2011)
4. Coombs, D., Herman, M., Hong, T.H., Nashman, M.: Real-Time Obstacle Avoidance Using Central Flow Divergence, and Peripheral Flow. IEEE Transactions on Robotics and Automation **14**(1), 49–59 (1998)
5. Engel, J., Schöps, T., Cremers, D.: LSD-SLAM: Large-Scale Direct monocular SLAM. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) European Conference on Computer Vision (ECCV). vol. 8690 LNCS, pp. 834–849. Springer International Publishing, Cham (2014)
6. Fanello, S.R., Rhemann, C., Tankovich, V., Kowdle, A., Escolano, S.O., Kim, D., Izadi, S.: HyperDepth: Learning Depth from Structured Light without Matching. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5441–5450. IEEE, Las Vegas, NV, USA (Jun 2016)
7. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient Belief Propagation for Early Vision. International Journal of Computer Vision (IJCV) **70**(1), 41–54 (Oct 2006)
8. Flacco, F., Kroger, T., De Luca, A., Khatib, O.: A depth space approach to human-robot collision avoidance. In: IEEE International Conference on Robotics and Automation (ICRA). pp. 338–345. IEEE, Saint Paul, MN (May 2012)
9. Furukawa, R., Sagawa, R., Kawasaki, H.: Depth Estimation Using Structured Light Flow — Analysis of Projected Pattern Flow on an Object's Surface. In: IEEE International Conference on Computer Vision (ICCV). pp. 4650–4658. IEEE (Oct 2017)
10. Gorthi, S.S., Rastogi, P.: Fringe projection techniques: Whither we are? Optics and Lasers in Engineering **48**(2), 133–140 (Feb 2010)
11. Green, W.E., Oh, P.Y.: Optic-Flow-Based Collision Avoidance. IEEE Robotics & Automation Magazine **15**(1), 96–103 (Mar 2008)
12. Grewal, H., Matthews, A., Tea, R., George, K.: LIDAR-based autonomous wheelchair. In: IEEE Sensors Applications Symposium (SAS). pp. 1–6. IEEE, Glassboro, NJ, USA (2017)
13. Gupta, M., Nayar, S.K.: Micro Phase Shifting. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 813–820. IEEE, Providence, RI (Jun 2012)
14. Gupta, M., Yin, Q., Nayar, S.K.: Structured Light in Sunlight. In: IEEE International Conference on Computer Vision (ICCV). pp. 545–552. IEEE, Sydney, Australia (Dec 2013)
15. Hansard, M., Lee, S., Choi, O., Horaud, R.: Time of Flight Cameras: Principles, Methods, and Applications. Springer Publishing Company, Incorporated (2012)
16. Heide, F., Heidrich, W., Hullin, M., Wetzstein, G.: Doppler time-of-flight imaging. ACM Transactions on Graphics **34**(4), 1–11 (Jul 2015)
17. Heinzmann, J., Zelinsky, A.: Quantitative Safety Guarantees for Physical Human-Robot Interaction. The International Journal of Robotics Research **22**(7-8), 479–504 (2003)
18. Hirschmuller, H.: Accurate and Efficient Stereo Processing by Semi-Global Matching and Mutual Information. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). vol. 2, pp. 807–814. IEEE, San Diego, CA, USA (2005)

19. Horn, B., Fang, Y.F.Y., Masaki, I.: Time to Contact Relative to a Planar Surface. In: IEEE Intelligent Vehicles Symposium. pp. 68–74. IEEE (2007)
20. Ikuta, K., Ishii, H., Nokata, M.: Safety Evaluation Method of Design and Control for Human-Care Robots. In: International Symposium on Micromechatronics and Human Science (MHS). pp. 119–127. IEEE (2000)
21. Kawasaki, H., Furukawa, R., Sagawa, R., Yagi, Y.: Dynamic scene shape reconstruction using a single structured light pattern. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1–8. IEEE, Anchorage, AK, USA (Jun 2008)
22. Lacevic, B., Rocco, P.: Kinetostatic danger field - a novel safety assessment for human-robot interaction. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 2169–2174. IEEE, Taipei (Oct 2010)
23. Li Zhang, Curless, B., Seitz, S.: Rapid shape acquisition using color structured light and multi-pass dynamic programming. In: 3D Data Processing Visualization and Transmission. pp. 24–36. IEEE Comput. Soc (2002)
24. Lin, J.F., Su, X.Y.: Two-dimensional Fourier transform profilometry for the automatic measurement of three-dimensional object shapes. Optical Engineering **34**(11), 3297 (Nov 1995)
25. Liu, C., Tomizuka, M.: Algorithmic safety measures for intelligent industrial co-robots. In: IEEE International Conference on Robotics and Automation (ICRA). pp. 3095–3102. IEEE, Stockholm, Sweden (May 2016)
26. Lucas, B.D., Kanade, T.: An Iterative Image Registration Technique with an Application to Stereo Vision. In: International Joint Conference on Artificial Intelligence (IJCAI). pp. 674–679. Vancouver, British Columbia, Canada (1981)
27. Marvel, J.A.: Performance Metrics of Speed and Separation Monitoring in Shared Workspaces. IEEE Transactions on Automation Science and Engineering **10**(2), 405–414 (Apr 2013)
28. Muller, D., Pauli, J., Nunn, C., Gormer, S., Muller-Schneiders, S.: Time to contact estimation using interest points. In: International IEEE Conference on Intelligent Transportation Systems (ITSC). pp. 1–6. IEEE, St. Louis (Oct 2009)
29. Mur-Artal, R., Tardos, J.D.: ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. IEEE Transactions on Robotics **33**(5), 1255–1262 (Oct 2017)
30. Nayar, S.K., Krishnan, G., Grossberg, M.D., Raskar, R.: Fast Separation of Direct and Global Components of a Scene using High Frequency Illumination. ACM Transactions on Graphics **25**(3), 935–944 (2006)
31. O'Toole, M., Achar, S., Narasimhan, S.G., Kutulakos, K.N.: Homogeneous codes for energy-efficient illumination and imaging. ACM Transactions on Graphics **34**(4), 1–13 (Jul 2015)
32. O'Toole, M., Mather, J., Kutulakos, K.N.: 3D Shape and Indirect Appearance by Structured Light Transport. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3246–3253. IEEE (2014)
33. Pagès, J., Salvi, J., Collewet, C., Forest, J.: Optimised De Bruijn patterns for one-shot shape acquisition. Image and Vision Computing **23**(8), 707–720 (Aug 2005)
34. Sagawa, R., Ota, Y., Yagi, Y., Furukawa, R., Asada, N., Kawasaki, H.: Dense 3D reconstruction method using a single pattern for fast moving object. In: IEEE International Conference on Computer Vision (ICCV). pp. 1779–1786. IEEE (Sep 2009)
35. Se Dong Min, Jin Kwon Kim, Hang Sik Shin, Yong Hyeon Yun, Chung Keun Lee, Myoungho Lee: Noncontact Respiration Rate Measurement System Using an Ultrasonic Proximity Sensor. IEEE Sensors Journal **10**(11), 1732–1739 (Nov 2010)
36. Takeda, M., Ina, H., Kobayashi, S.: Fourier-transform method of fringe-pattern analysis for computer-based topography and interferometry. Journal of the Optical Society of America **72**(1), 156 (Jan 1982)

37. Takeda, M., Mutoh, K.: Fourier transform profilometry for the automatic measurement of 3-D object shapes. Applied Optics **22**(24), 3977 (Dec 1983)
38. Van der Jeught, S., Dirckx, J.J.: Real-time structured light profilometry: A review. Optics and Lasers in Engineering **87**, 18–31 (Dec 2016)
39. Vo, M., Narasimhan, S.G., Sheikh, Y.: Separating Texture and Illumination for Single-Shot Structured Light Reconstruction. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 433–440. IEEE (Jun 2014)
40. Wang, J., Bartels, J., Whittaker, W., Sankaranarayanan, A.C., Narasimhan, S.G.: Programmable Triangulation Light Curtains. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) European Conference on Computer Vision (ECCV). vol. 11207, pp. 20–35. Springer International Publishing, Cham (2018)
41. Watanabe, Y., Sakaue, F., Sato, J.: Time-to-contact from image intensity. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4176–4183. IEEE, Boston, MA, USA (Jun 2015)
42. Yang, Q., Wang, L., Ahuja, N.: A constant-space belief propagation algorithm for stereo matching. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1458–1465. IEEE, San Francisco, CA, USA (Jun 2010)
43. Zhang, Z.: Microsoft Kinect Sensor and Its Effect. IEEE Multimedia **19**(2), 4–10 (Feb 2012)
44. Zinn, M., Khatib, O., Roth, B., Salisbury, J.: Playing it safe. IEEE Robotics & Automation Magazine **11**(2), 12–21 (Jun 2004)