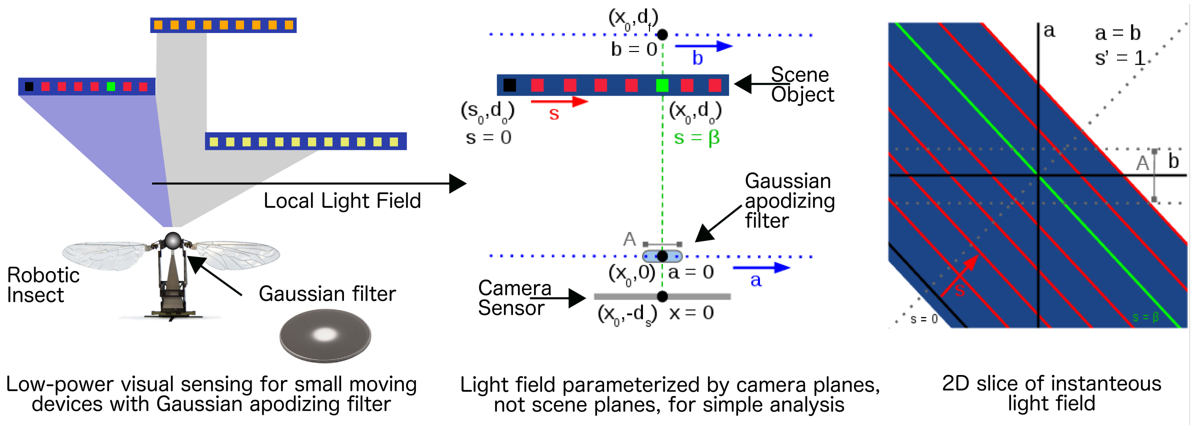
Low Power Depth and Velocity from a Passive Moving Sensor

Emma Alexander
Harvard SEAS
Cambridge, MA
ealexander@seas.harvard.edu

Sanjeev J. Koppal
University of Florida
Gainesville, FL
sjkoppal@ece.ufl.edu

Todd Zickler
Harvard SEAS
Cambridge, MA
zickler@seas.harvard.edu



Low-power visual sensing for small moving devices with Gaussian apodizing filter

Light field parameterized by camera planes, not scene planes, for simple analysis

2D slice of instanteous light field

Figure 1: Based on the analysis of a view-parameterized light field, a passive, monocular sensor could instantaneously measure distances and velocities of objects from image derivatives in a window, even under extreme power constraints.

Abstract

We present an opportunity for the visual sensing of depth and 3D velocity using a passive sensor that has extremely low power requirements. This opportunity comes from a new mathematical constraint, which we derive, that relates depth and velocity to spatial and temporal derivatives of image values captured by a coded-aperture camera that observes a moving scene. The constraint exploits the fact that there are two causes of brightness change in this situation: features move across the image due to motion, and contrast changes because of time-varying optical blur. The sensor that could be realized from this constraint is called a focal flow sensor. We analytically characterize the working volume of such a sensor in relation to its size, and we provide simulation results that affirm its viability.

1. Extreme Power Constraints

The miniaturization of technology is constantly advancing, and platforms such as tiny air vehicles increasingly demand visual sensors that operate on smaller scales and with less power than current technology can achieve [1, 6]. One way to reduce power requirements in these situations is through computational sensing, where optics and inference algorithms are co-designed in ways that lessen the complexity of post-capture calculations. This paper presents mathematical analysis that suggests a new type of computational sensor, one that measures distance to visible surfaces and 3D velocity relative to those surfaces. This could provide a low-power alternative to existing, high-power depth sensors that either require an active light source (e.g. time-of-flight) or substantial post-capture computation to solve a complex inference problem (e.g. stereo, depth from defocus).

The cues studied here are motion and defocus. Deriving depth from either of these signals independently can be expensive or unreliable, but their weaknesses can be mitigated through a novel cue combination mechanism. Our contribution is the derivation of a per-pixel constraint,

$$\boxed{\begin{bmatrix} I_y & I_x & xI_x + yI_y & I_{xx} + I_{yy} \end{bmatrix} \cdot \vec{v} + I_t \approx 0,}$$

which holds when the aperture of a moving camera is equipped with an apodizing filter that has a narrow Gaussian profile. Over an image patch, depth and velocity are recovered simply by taking spatial and temporal derivatives, and solving a 4×4 linear system for coefficients $\vec{v} = (v_0, v_1, v_2, v_3)$. Scene parameters are then computed from these coefficients in closed form using known intrinsic camera parameters such as aperture size and focal length.

The proposed sensor can be understood as an optical flow sensor with defocus. Traditional optical flow, where all images are in focus, is computable from a linear system of equations on image derivatives in a receptive field [7]. This locally resolves time-to-contact [5], but it cannot instantaneously give explicit local distance and velocity. We show that when the camera is defocused and equipped with an appropriate aperture function, a similarly simple calculation provides explicit depth and velocity. The derivations of our results appear in the supplemental material, and they will be published at a later time.

2. View-Parameterized Light Field

We derive our constraint using a light field representation [2, 11], which has successfully been used to analyze a variety of computational cameras [9, 10]. For clarity, we present in two dimensions (flatland), where there is a 2D space of light rays and a 1D image domain. As is common, we parameterize light rays by their intersections with two parallel reference lines, but less common, we affix the reference lines to the moving sensor. This makes a view-parameterized light field. The references are parallel to the photodetector line and located at the lens center and object-side focal point, all of which are determined by the sensor's internal geometry and assumed known (middle of Figure 1).

We assume that locally the scene is fronto-parallel and of matte reflectance, as shown in Figure 1. There are four parallel lines of interest in the figure center: the local world line parameterized by s, the photodetector (image) line parameterized by x, the first reference line at the lens center parameterized by a, and the second reference line at the object-side focal point parameterized by b. We use the word *texture* for the radiance at the world line and denote it $T(s)$, and we assume that it is at least twice differentiable.

The axial distances from the lens center to the sensor (μ_s) and focal point (μ_f) are known quantities determined by the sensor's construction, while the distance to the world plane, or depth, (Z) is to be measured. The origins of the lens, focal, and sensor lines are at their intersection with the optical axis. The world line has its own origin, so its intersection with the optical axis is at world point $s - X$, with X the (unknown) time-varying lateral position of the sensor. Our aim is to recover the depth Z and the sensor velocity $(\dot{X}, \dot{Z})$ from image measurements, and to do so in a way that is invariant to the unknown texture $T(s)$.

Each texture point induces a ray in the light field, and the slope of these rays encodes depth. The light rays rotate about their intersection with the line $a = b$ in response to axial motion $\dot{Z}$, and in response to transverse motion $\dot{X}$ they translate along the line $a = b$. We are interested in the radiance $L(a, b)$ of the light ray that corresponds to a fixed world point. This world point projects to a time-varying image location $x(t)$, and its radiance is determined by where it intersects the texture plane:

$$L(a,b,T,X,Z) = T(s(a,b(x(t)),X(t),Z(t)). \tag{1}$$

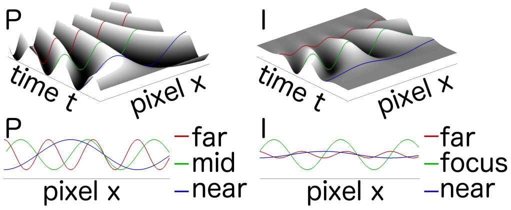


Figure 2: Time-varying images of a 1D front-parallel texture with a sinusoidal radiance pattern. *Left*: In the all-in-focus case, there is no contrast loss, and the image $P(x,t)$ changes only in frequency and phase. *Right*: A finite aperture incurs optical blur, and now the contrast of image $I(x,t)$ also changes over time, allowing explicit recovery of depth and velocity without knowledge of texture.

3. Conventional All-in-Focus Constraint

Before proceeding to the focal flow constraint, it is worth understanding how the classical linearized optical flow (or constant brightness) equation [5] can be derived in this light field framework. At time t, the all-in-focus (pinhole) image $P(x,t)$ corresponds to a slice through the light field along line $a = 0$:

$$P(x(t),t) = L(0, b(x(t)), T, X(t), Z(t)). \tag{2}$$

As the sensor moves through a static scene, the effective light field skews, and the image changes. The left of Figure 2 is an example where the texture $T(s)$ is sinusoidal and the velocity is zero in X and constant in Z. Because there is no optical blur, there is no loss of contrast over time, and the imaged sinusoid changes only in frequency and phase.

The linearized optical flow constraint follows directly from taking the total time derivative of Eq. 2 and noting that, because of the fixed contrast, $dP/dt = 0$. Alternatively, the partial image derivatives can be rearranged in the form of a related linear constraint on time-to-contact ($Z/\dot{Z}$) and bearing ($\dot{Z}/\dot{X}$):

$$\begin{bmatrix} P_x & xP_x \end{bmatrix} \cdot \begin{bmatrix} v_1 & v_2 \end{bmatrix} + P_t = 0, \tag{3}$$

$$\text{time-to-contact: } 1/v_2 \tag{4}$$

$$\text{bearing: } -\mu_s v_2 / v_1. \tag{5}$$

Our derivation follows directly from the differentiability of the texture $T(s)$ and light field $L(a, b(t))$. It is an alternative to previous derivations based on a truncated Taylor expansion of the image [3].

The linear constraint of Eq. 3 holds at every pixel, so image derivatives from a small (non-degenerate) image patch can be accumulated into a simple 2×2 linear system that uniquely determines bearing and time-to-contact. However, there is not enough information to resolve this into explicit depth and velocity.

4. Focal Flow Constraint

The view-parameterized light field makes it easy to add an aperture. We give our sensor a finite aperture that passes all rays through $a \in [-A/2, A/2]$, and in the spirit of coded aperture cameras [8, 12, 13] we include an attenuating transmittance function $k(a)$. In this case, the (possibly defocused) image is

$$I(x(t),t) = \int_{-A/2}^{A/2} L(a, b(x(t)), T, X(t), Z(t)) k(a) da. \tag{6}$$

As depicted in the right of Figure 1, at each pixel x this is a vertical line integral of the light field, weighted by the aperture function.

Now, when the sensor moves, changes in optical blur result in changing image contrast over time. The right of Figure 2 shows this effect for the special case of a sinusoidal texture pattern. It is this change in contrast, which comes in addition to the changes in frequency and phase, that provides additional information to resolve time-to-contact and bearing into explicit depth Z and velocity $(\dot{X}, \dot{Z})$.

Note that unlike the pinhole case, the total time derivative of Eq. 6 is not zero. Instead, it takes on a value $\frac{dI}{dt} = E(T)$ that depends on the unknown texture pattern T. However, it can be shown that for a suitable choice of the aperture function $k(a)$ this 'error term' is directly proportional to a very measurable quantity: the second spatial derivative of the image, $E(T) \propto I_{xx}$. The required function is a truncated Gaussian,

$$k(a) = \begin{cases} e^{-\frac{a^2}{2\Sigma^2}} & , \ |a| \le A/2 \\ 0 & , \ |a| > A/2 \end{cases}, \tag{7}$$

whose width is sufficiently narrow with respect to the aperture (say, $\Sigma < A/6$).

The ratio between the image derivative I_{xx} and error term E contains depth and velocity information and can be estimated from the image alone, and this leads to the following texture-independent constraint on depth and velocity.

$$\begin{bmatrix} I_x & xI_x & I_{xx} \end{bmatrix} \cdot \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix} + I_t = 0, \tag{8}$$

$$Z = \frac{\mu_s^2 \Sigma^2 v_2}{\mu_s^2 \Sigma^2 v_2 / \mu_f - \mu_f v_3} \tag{9}$$

$$\dot{Z} = -Zv_2 \tag{10}$$

$$\dot{X} = Zv_1/\mu_s \tag{11}$$

This per-pixel linear constraint can be applied to a small image patch for power-efficient estimates of depth and velocity. The analogous constraint on two-dimensional textures that is shown in the introduction follows immediately from the separability of the Gaussian aperture. In this case, the other component of lateral velocity is $\dot{Y} = Zv_0/\mu_s$.

5. Working Range

There are many algorithmic choices for a sensor using our constraint, such as the scale of image derivatives and the grouping of pixels into appropriate patches. These are longstanding questions in optical flow and time to contact [4, 5] and instead of addressing them here we study the underlying sensitivity of the system by considering observations of a sinusoidal texture. From these idealized images we can derive bounds on depth error for sensors that vary in aperture size and other physical dimensions, and we can visualize how working range relates to sensor size.

When a moving camera observes a sinusoidal texture it obtains sinusoidal images with frequencies $\omega(t)$ and amplitudes $B(t)$. In this context, we can analytically derive an upper bound on depth error by propagation of errors in measured frequency (e.g. due to spatial resolution) and of errors in measured amplitude (e.g. due to bit-depth and sensor noise). If image frequencies and their changes are measured with error less than ϵ_ω and $\epsilon_{\dot{\omega}}$, and image brightnesses and their changes within ϵ_B and $\epsilon_{\dot{B}}$, respectively, then the error in estimated depth ϵ_Z is bounded as:

$$\begin{aligned} \epsilon_Z \le & \sqrt{\left(\frac{\partial Z}{\partial \omega}\right)^2 \epsilon_\omega^2 + \left(\frac{\partial Z}{\partial \dot{\omega}}\right)^2 \epsilon_{\dot{\omega}^2} + \left(\frac{\partial Z}{\partial B}\right)^2 \epsilon_B^2 + \left(\frac{\partial Z}{\partial \dot{B}}\right)^2 \epsilon_{\dot{B}}^2} \\ = & \frac{Z}{\mu_f}(Z - \mu_f) \sqrt{\frac{\epsilon_\omega^2}{\omega^2} + \frac{\epsilon_{\dot{\omega}}^2}{\dot{\omega}^2} + \frac{\epsilon_B^2}{B^2} + \frac{\epsilon_{\dot{B}}^2}{\dot{B}^2}}. \end{aligned} \tag{12}$$

This error bound is shown in the left of Figure 3 for sensors with the same aperture size but different distances μ_s. For each sensor we plot the bounded errors for textures located at distances Z within a $20cm$ window around the object-side focal point. These graphs agree with the intuition that the strength of the blur cue diminishes when the texture moves too far from the focal point (recall the right of Figure 2). The spike in the error bound at the focal point μ_f is caused by the $1/\dot{B}$ term in expression (12), and it reveals the limitations of a first-order propagation-of-errors: in simulations of actual depth reconstructions, we do not see such errors near the focal point. Note that the appearance of both frequency and brightness error terms in expression (12) reveals a trade-off between spatial resolution and bit-depth. For a desired level of depth accuracy, a camera with high bit-depth (low ϵ_B and $\epsilon_{\dot{B}}$) or high pixel density (low ϵ_ω and $\epsilon_{\dot{\omega}}$) could make up for deficiencies in the other.

One can draw similar error graphs for different aperture sizes A, and for each combination of μ_s and A we can compute an ϵ-working range, defined as the range of positions Z for which the sensor's depth error is guaranteed to be less than ϵ. The right of Figure 3 shows one such graph for $\epsilon = 0.25cm$. This visualization can be used to identify optimal combinations of sensor length and width (μ_s and A) in the face of constraints on fixed total area $\mu_s \times A$.

We also verify the viability of this sensing method by simulating noisy images of sinusoidal plaid textures, approximating derivatives by finite differences, accumulating per-pixel constraints over a 50×50 window, and recovering depth using Eq. 9. We do this for sensors of various dimensions, with texture frequencies adjusted so that every sensor captures the same image when the world plane is at its focal point. Figure 4 shows such distance estimates averaged over 50 trials, for sensors having the same aperture size but different lengths μ_s. Accuracy is higher near each sensor's focal point and degrades gradually over its working range.

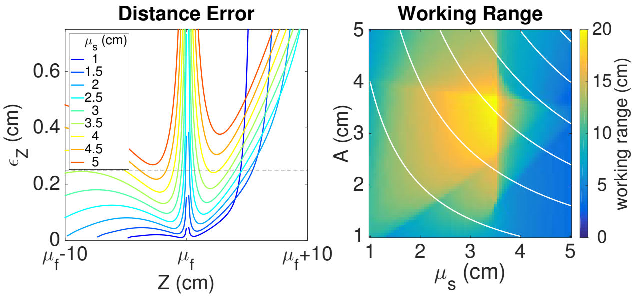


Figure 3: An exploration of noise sensitivity with $\epsilon_B = \epsilon_{\dot{B}}/2 = \epsilon_\omega = \epsilon_{\dot{\omega}}/2 = .05$, $\dot{d}o = 1$, and $\mu_f = 5\mu_s$, with a sinusoidal texture of unit frequency in world coordinates. *Left*: for $A = 3$, distance error ϵ_Z is shown over μ_s as a function of distance, shifted to align each camera's focal point. Dotted line at $0.25cm$ marks threshold defining the 0.25-working range. *Right*: 0.25-working range for varying camera dimensions. In white are level curves of camera area $A \times \mu_s$, increasing by $4cm^2$ from bottom left.

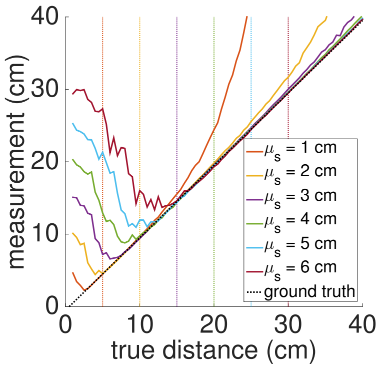


Figure 4: Simulated cameras of different size (A = 3 cm, $\Sigma = A/6$ for all) show working ranges of varying size and location. Near each camera's focal distance, indicated by a dashed vertical line of corresponding color, measurements closely match ground truth.

6. Toward a Focal Flow Camera

To realize a focal flow sensor, we are currently exploring robust estimation techniques that compute derivatives at multiple spatial scales and that automatically discard image windows that to not contain sufficient brightness variation or do not back-project to fronto-planar scene planes. We are also testing physical prototypes and exploring the mathematical space of apodizing functions that provide texture-invariance in ways similar to the truncated Gaussian. More generally, we believe that the view-parameterized light field may be useful in modeling and designing other computational sensors that exploit various optical cues.

Acknowledgements This material is based upon work supported by Texas Instruments Incorporated and the National Science Foundation Graduate Research Fellowship under Grant No. (NSF grant number).

References

[1] D. Floreano, J.-C. Zufferey, M. V. Srinivasan, and C. Ellington. *Flying insects and robots*. Springer, 2009.

[2] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The lumigraph. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 43–54. ACM, 1996.

[3] B. K. Horn, Y. Fang, and I. Masaki. Time to contact relative to a planar surface. In *Intelligent Vehicles Symposium, 2007 IEEE*, pages 68–74. IEEE, 2007.

[4] B. K. Horn, Y. Fang, and I. Masaki. Hierarchical framework for direct gradient-based time-to-contact estimation. In *Intelligent Vehicles Symposium, 2009 IEEE*, pages 1394–1400. IEEE, 2009.

[5] B. K. Horn and B. G. Schunck. Determining optical flow. In *1981 Technical Symposium East*, pages 319–331. International Society for Optics and Photonics, 1981.

[6] S. J. Koppal, I. Gkioulekas, T. Zickler, and G. L. Barrows. Wide-angle micro sensors for vision on a tight budget. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 361–368. IEEE, 2011.

[7] D. N. Lee. A theory of visual control of braking based on information about time-to-collision. *Perception*, (5):437–59, 1976.

[8] A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Image and depth from a conventional camera with a coded aperture. In *ACM Transactions on Graphics (TOG)*, volume 26, page 70. ACM, 2007.

[9] A. Levin, W. T. Freeman, and F. Durand. Understanding camera trade-offs through a bayesian analysis of light field projections. In *Computer Vision–ECCV 2008*, pages 88–101. Springer, 2008.

[10] A. Levin, S. W. Hasinoff, P. Green, F. Durand, and W. T. Freeman. 4d frequency analysis of computational cameras for depth of field extension. In *ACM Transactions on Graphics (TOG)*, volume 28, page 97. ACM, 2009.

[11] M. Levoy and P. Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42. ACM, 1996.

[12] A. Veeraraghavan, R. Raskar, A. Agrawal, A. Mohan, and J. Tumblin. Dappled photography: Mask enhanced cameras for heterodyned light fields and coded aperture refocusing. *ACM Trans. Graph.*, 26(3):69, 2007.

[13] C. Zhou, S. Lin, and S. Nayar. Coded aperture pairs for depth from defocus. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 325–332. IEEE, 2009.

Appendices

A. Constraint for Conventional Sensor

We assume a pinhole camera images a 1-D Lambertian fronto-parallel planar object at distance $Z(t)$ with unchanging texture $T(s)$.

Because the pinhole camera captures images that are in focus everywhere, it does not have a well-defined focal plane. To parametrize L as discussed previously we fix an arbitrary plane parallel to the sensor and take μ_f to be its constant distance from the pinhole.

Letting $x(t)$ refer to the changing image location corresponding to a fixed world point, we see from similar triangles that

$$b(x(t)) = -\frac{\mu_f}{\mu_s} x(t). \tag{13}$$

Finding the slope m of the line generated in the light field by this point can be found by requiring a single ray of light to pass through scene points $(a, 0)$, $(s - X(t), Z(t))$, and (b, μ_f). Requiring the physical slopes between these points to be equal characterizes the resulting line in the light field:

$$\begin{aligned} a =& m(t)b(x(t)) + (1 - m(t))(s - X(t)), && (14) \\ m(t) =& \frac{Z(t)}{Z(t) - \mu_f}. && (15) \end{aligned}$$

From equation (14), note the following expression for texture parameter s.

$$s(a, b, m(t), X(t)) = \frac{a - m(t)b(x(t))}{1 - m(t)} + X(t) \tag{16}$$

We can now express the light field as a function not only of a and b but also of a specific texture and scene geometry, as below.

$$L(a, b, T, m(t), X(t)) = T(s(a, b, m(t), X(t))) \tag{17}$$

A pinhole camera blocks all light that does not pass through $a = 0$, so it samples an image along the a axis of the light field as in equation (2).

We examine the total time derivative of the image under the assumption of unchanging texture:

$$\begin{aligned} \frac{dP}{dt} =& \frac{\partial P}{\partial x}\frac{dx}{dt} + \frac{\partial P}{\partial t} && (18) \\ =& \frac{dL}{dt} = \frac{dT}{dt} = 0. && (19) \end{aligned}$$

A closer look at the time derivative of the texture T produces an expression that depends explicitly on texture and scene geometry. Equations (14) through (16) and (13), along with their derivatives, are used between lines 2 and 5 below.

$$\begin{aligned} \frac{dT}{dt} =& \frac{d}{dt} T(s(0, b(x(t)), m(t), X(t))) && (20) \\ =& \frac{\partial T}{\partial s}\left(\frac{\partial s}{\partial b}\frac{db}{dx}\frac{dx}{dt} + \frac{\partial s}{\partial m}\frac{dm}{dt} + \frac{\partial s}{\partial X}\frac{dX}{dt}\right) && (21) \\ =& \frac{\partial T}{\partial s}\left(\frac{-m}{1-m}\frac{-\mu_f}{\mu_s}\frac{dx}{dt} + \frac{(a-b)}{(1-m)^2}\frac{-\mu_f \dot{Z}}{(Z-\mu_f)^2} + \frac{dX}{dt}\right) && (22) \\ =& \frac{\partial T}{\partial s}\left(\frac{Z}{-\mu_s}\frac{dx}{dt} + \frac{(0 + x\mu_f/\mu_s)}{\mu_f^2/(Z-\mu_f)^2}\frac{-\mu_f \dot{Z}}{(Z-\mu_f)^2} + \frac{dX}{dt}\right) && (23) \\ =& \left(\frac{\partial T}{\partial s}\frac{Z}{-\mu_s}\right)\frac{dx}{dt} + \left(\frac{\partial T}{\partial s}\right)\left(\frac{x}{-\mu_s}\frac{dZ}{dt} + \frac{dX}{dt}\right) && (24) \end{aligned}$$

Comparing expression (24) to equation (18) gives an explicit form for image derivatives.

$$\begin{aligned} \frac{\partial P}{\partial x} =& \left(\frac{\partial T}{\partial s}\right)\frac{Z}{-\mu_s} && (25) \\ \frac{\partial P}{\partial t} =& \left(\frac{\partial T}{\partial s}\right)\left(\frac{x}{-\mu_s}\frac{dZ}{dt} + \frac{dX}{dt}\right) && (26) \end{aligned}$$

Note that the ratio of $\frac{\partial P}{\partial x}$ and $\frac{\partial P}{\partial t}$ is independent of texture and contains only information about known pixel location x and camera size μ_s and unknown depth Z and camera velocity $(\frac{dX}{dt}, \frac{dZ}{dt})$. This allows us to separate equations (25) and (26) into image and scene information and rewrite them in the form seen in equation (3), with texture-independent coefficients v_1 and v_2.

$$\begin{aligned} v_1 =& \mu_s \frac{\dot{X}}{Z} && (27) \\ v_2 =& -\frac{\dot{Z}}{Z} && (28) \end{aligned}$$

These coefficient contain the information needed to calculate time-to-contact and direction of travel. Methods for computing these quantities robustly from image data are a topic of ongoing research [4]. Our primary contribution lies in the comparison between these equations (3-5) and the analogous expressions (8-11) for the blurred case, shown below.

B. Constraint for Focal Flow Sensor

We now assume a camera with an ideal thin lens and coded aperture images the 1-D Lambertian fronto-parallel planar object at distance $Z(t)$ with unchanging texture $T(s)$.

Replacing the pinhole with a lens means that at each pixel the camera integrates over a line of a values rather than taking an infinitesimally thin sample of the light field. When an object is not in focus, texture lines in the light field have a nonzero slope and this integration results in blur, leading to an error term in the optical flow equation. The essential mathematical difference is that the a term in equation (22) no longer disappears. We show that with a coded aperture of Gaussian transmittance, this error term can be used as data that enables direct measurement of depth and camera velocity, as shown below.

For a lens centered at $a = 0$ of aperture size A and aperture function $k(a)$, the image is formed by the weighted integration of the light passing through the lens to reach each pixel, as in equation (6). At this point we make no assumptions about $k(a)$, though in a standard lens it would be a box function in the shape of the aperture. Note that if we were to express k in terms of x it would vary based on depth, but in light field coordinates it remains fixed and depth dependence operates through the changing slope of the texture lines.

Because each image is blurred, intensity changes now result from both feature motion and changes in defocus level, so the unchanging-texture constraint no longer guarantees an unchanging image and $\frac{dI}{dt}$ is not identically zero. Instead, we propose a modification to the optical flow equation for cameras with significant apertures. This change is specified by the total time derivative of the light field integral in equation (6), explored below.

$$\frac{dI}{dt} = \frac{d}{dt}\int_{-A/2}^{A/2} L(a, b(x(t)), T(s), m(t), X(t))k(a)da \tag{29}$$

$$= \int_{-A/2}^{A/2} \left(\frac{\partial L}{\partial b}\frac{db}{dx}\frac{dx}{dt} + \frac{\partial L}{\partial m}\frac{dm}{dt} + \frac{\partial L}{\partial X}\frac{dX}{dt}\right)k(a)da \tag{30}$$

$$\begin{aligned}= &\left(\int_{-A/2}^{A/2} \frac{\partial L}{\partial b}\frac{db}{dx}k(a)da\right)\frac{dx}{dt} + \\ &\int_{-A/2}^{A/2} \left(\frac{\partial L}{\partial m}\frac{dm}{dt} + \frac{\partial L}{\partial X}\frac{dX}{dt}\right)k(a)da\end{aligned} \tag{31}$$

As before, comparing expressions (18) and (31) reveals the form of image derivatives, while equations (14) through (16), (13), and their derivatives give their explicit values.

$$\frac{\partial I}{\partial x} = \int_{-A/2}^{A/2} \frac{\partial L}{\partial b}\frac{db}{dx}k(a)da \tag{32}$$

$$= \frac{Z}{-\mu_s}\int_{-A/2}^{A/2} k(a)\frac{\partial T}{\partial s}da \tag{33}$$

$$\frac{\partial I}{\partial t} = \int_{-A/2}^{A/2} \left(\frac{\partial L}{\partial m}\frac{dm}{dt} + \frac{\partial L}{\partial X}\frac{dX}{dt}\right)k(a)da \tag{34}$$

$$\begin{aligned}&= \left(\frac{\dot{Z}}{-\mu_s}x + \dot{X}\right)\int_{-A/2}^{A/2} k(a)\frac{\partial T}{\partial s}da \\ &\quad + -\frac{\dot{Z}}{\mu_f}\int_{-A/2}^{A/2} ak(a)\frac{\partial T}{\partial s}da\end{aligned} \tag{35}$$

As aperture size A approaches zero, these match equations (25) and (26) found in the blur-free case. The second term in $\frac{\partial I}{\partial t}$, however, is significant because unlike equations (25) and (26), the ratio of (34) and (35) is not independent of texture and so cannot be used in traditional calculations for time-to-contact and camera direction without introducing a texture-dependent, blur-induced error term.

We now consider the case that the aperture function is a narrow truncated Gaussian. This can be accomplished with the addition of a mask whose transmittance along a obeys the equation (7). With this filter, the second term in equation (35), which we will call E, can be analyzed. We break it into two terms, one from the Gaussian term and one from the truncation.

$$E = -\frac{\dot{Z}}{\mu_f}\int_{-A/2}^{A/2} ak(a)\frac{\partial T}{\partial s}da \tag{36}$$

$$= E_0 + E_1, \tag{37}$$

$$E_0 = -\frac{\dot{Z}}{\mu_f}\int_{-\infty}^{\infty} ak(a)\frac{\partial T}{\partial s}da, \tag{38}$$

$$\begin{aligned}E_1 = &\frac{\dot{Z}}{\mu_f}\int_{-\infty}^{-A/2} ak(a)\frac{\partial T}{\partial s}da \\ &+ \frac{\dot{Z}}{\mu_f}\int_{A/2}^{\infty} ak(a)\frac{\partial T}{\partial s}da.\end{aligned} \tag{39}$$

The second term, E_1, is a correction term that distinguishes our case from an aperture of infinite extent. This should almost always be negligible for a narrow Gaussian. For instance, if $\Sigma = A/6$ then the regions included in E_1 contains less than 2% of the total weight of $ak(a)$. For even the most pathological textures, only a very small number of pixels could produce a value for E_1 on the order of E_0.

The first term E_0 can be expanded as follows with partial integration. Note that $\frac{\partial a}{\partial s} = (1 - m)$ by equation (14) and that for Gaussian $k(a) = e^{-a^2/2\Sigma^2}$, $\frac{dk}{da} = -\frac{a}{\Sigma^2}k(a)$.

$$E_0 = -\frac{\dot{Z}}{\mu_f}\int_{-\infty}^{\infty} ak(a)\frac{\partial T}{\partial s}da \tag{40}$$

$$= -\frac{\dot{Z}}{\mu_f}\int_{-\infty}^{\infty} \left(-\Sigma^2\frac{\partial k}{\partial a}\right)\left(\frac{\partial T}{\partial a}\frac{\partial a}{\partial s}\right)da \tag{41}$$

$$= -\frac{\dot{Z}}{\mu_f}\Sigma^2(m-1)\int_{-\infty}^{\infty} \frac{\partial k}{\partial a}\frac{\partial T}{\partial a}da \tag{42}$$

$$
\begin{aligned}
&= -\frac{\dot{Z}}{\mu_f}\Sigma^2(m-1) \\
&\left(\left[k(a)\frac{\partial T}{\partial a}\right]_{-\infty}^{\infty} - \int_{-\infty}^{\infty} k(a)\frac{\partial^2 T}{\partial a^2}da\right)
\end{aligned} \tag{43}
$$

$$
=0 + \frac{\dot{Z}}{\mu_f}\Sigma^2(m-1)\int_{-\infty}^{\infty} k(a)\frac{\partial^2 T}{\partial a^2}da. \tag{44}
$$

$$
\tag{45}
$$

This term can be expressed in terms of the second texture derivative and a new camera constant c_E.

$$
c_E = \left(\frac{\mu_s \Sigma}{\mu_f}\right)^2 \tag{46}
$$

$$
E_0 = -\frac{\dot{Z}}{\mu_f}\frac{\Sigma^2}{m-1}\int_{-\infty}^{\infty} k(a)\frac{\partial^2 T}{\partial s^2}da \tag{47}
$$

$$
=\frac{\dot{Z}Zc_E}{\mu_s^2 m}\int_{-\infty}^{\infty} k(a)\frac{\partial^2 T}{\partial s^2}da \tag{48}
$$

This weighted, blurred second derivative of the texture is a quantity that we can easily connect to the second spatial derivative of the captured image. Note that the first line below is from equation (34) and the derivatives in line 2 are replaced by their values specified by equations (16) and (13) in line 3.

$$
\frac{\partial^2 I}{\partial x^2} = \frac{\partial}{\partial x}\left(\frac{Z}{-\mu_s}\int_{-A/2}^{A/2} k(a)\frac{\partial T}{\partial s}da\right) \tag{49}
$$

$$
=\frac{Z}{-\mu_s}\int_{-A/2}^{A/2} k(a)\left(\frac{\partial b}{\partial x}\frac{\partial s}{\partial b}\frac{\partial}{\partial s}\right)\frac{\partial T}{\partial s}da \tag{50}
$$

$$
=\left(\frac{Z}{-\mu_s}\right)^2\int_{-A/2}^{A/2} k(a)\frac{\partial^2 T}{\partial s^2}da \tag{51}
$$

$$
=\frac{Z}{\dot{Z}}\frac{m}{c_E}E_0 + E_2, \tag{52}
$$

$$
\begin{aligned}
E_2 = &-\frac{Z}{\dot{Z}}\frac{m}{c_E}\int_{-\infty}^{-A/2} k(a)\frac{\partial^2 T}{\partial s^2}da \\
&-\frac{Z}{\dot{Z}}\frac{m}{c_E}\int_{A/2}^{\infty} k(a)\frac{\partial^2 T}{\partial s^2}da.
\end{aligned} \tag{53}
$$

The coefficient on E_0 in the expression above is $-v_3$ as referenced in equation (8). Again, we can reasonably expect the error term E_2 to be negligible compared to E_0. The governing equation for the blurred case now takes the following form:

$$
v_1 I_x + v_2 x I_x + I_t = -v_3 I_{xx} + E_1 + E_2, \tag{54}
$$

$$
v_3 = -\frac{\dot{Z}}{Z}\frac{\mu_s^2\Sigma^2}{\mu_f^2 m}. \tag{55}
$$

When the filter is narrow so that E_1 and E_2 can be taken to vanish, equation (54) reduces to the relationship among measurable image quantities and scene unknowns stated in equation (8). In addition to v_1 and v_2 defined in equations (27) and (28), we have introduced the new texture-free, scene-dependent coefficient v_3. Because the ratio $v_3/v_2 = c_E/m$ isolates the depth-dependent light field slope m, this third scene coefficient provides a depth measurement. This leads to solutions to all scene geometry unknowns, shown in equations (9)-(11).

Compare this to the equation (3) in the unblurred case, which contains only v_1 and v_2. These quantities contain solutions for camera motion direction and time-to-contact, but without the blur-induced term v_3 the complete scene parameters cannot be deduced from image information.

B.1. 2D Focal Flow Constraint

In two dimensions, we begin with the requirement that a single ray passes through points $(a_x, a_y, 0)$, $(s_x - X, s_y - Y, Z)$, and (b_x, b_y, μ_f), which leads to the equations below. Note that, from the fronto-parallel planar assumption, $m(t)$ does not vary along x or y. All limits of integration are $-A/2$ to $A/2$. The filter is assumed narrow so that the error terms mirroring those in expression (54) are taken to be zero.

$$
T(s_x, s_y) = L(a_x, a_y, b_x, b_y, T, m(t), X(t), Y(t)) \tag{56}
$$

$$
s_x = \frac{a_x - m(t)b_x}{1 - m(t)} + X(t) \tag{57}
$$

$$
s_y = \frac{a_y - m(t)b_y}{1 - m(t)} + Y(t) \tag{58}
$$

$$
K = \frac{1}{2\pi\Sigma^2}e^{-\frac{a_x^2 + a_y^2}{2\Sigma^2}} = k(a_x)k(a_y) \tag{59}
$$

$$
I(x(t), y(t), t) = \iint LK da_x da_y \tag{60}
$$

$$
\frac{\partial I}{\partial x} = \frac{Z}{-\mu_s}\iint \frac{\partial T}{\partial s_x}k(a_x)k(a_y)da_x da_y \tag{61}
$$

$$
\frac{\partial I}{\partial y} = \frac{Z}{-\mu_s}\iint \frac{\partial T}{\partial s_y}k(a_x)k(a_y)da_x da_y \tag{62}
$$

$$
\begin{aligned}
\frac{\partial I}{\partial t} = &\left(\frac{\dot{Z}}{-\mu_s}x + \dot{X}\right)J_x + \left(\frac{\dot{Z}}{-\mu_s}y + \dot{Y}\right)J_y + \\
&\frac{\dot{m}}{(m-1)^2}\left(\iint \frac{\partial T}{\partial s_x}a_x K da_x da_y\right. \\
&\left. + \iint \frac{\partial T}{\partial s_y}a_y K da_x da_y\right),
\end{aligned} \tag{63}
$$

$$
J_x = \frac{-\mu_s}{Z}\frac{\partial I}{\partial x}, J_y = \frac{-\mu_s}{Z}\frac{\partial I}{\partial y} \tag{64}
$$

Because the aperture function is separable, as noted in equation (59), the final terms in $\frac{\partial I}{\partial t}$ can be resolved as before.

$$\frac{\dot{Z}}{\mu_f}\iint \frac{\partial T}{\partial s_x} a_x K = \frac{\dot{Z}}{\mu_f}\int \left[\int \frac{\partial T}{\partial s_x} a_x k(a_x) da_x\right] k(a_y) da_y \tag{65}$$

$$= \frac{\dot{Z}}{\mu_f}\int \left[\frac{\Sigma^2}{m-1}\int k(a_x)\frac{\partial^2 T}{\partial s_x^2}\right] k(a_y) da_y \tag{66}$$

$$= \frac{\dot{Z}}{\mu_f}\frac{\Sigma^2}{m-1}\iint \frac{\partial^2 T}{\partial s_x^2} K da_x da_y \tag{67}$$

$$= \frac{\dot{Z} c_E}{Zm}\frac{\partial^2 I}{\partial x^2} \tag{68}$$

This leads to the following alteration of equation (8) to account for the additional spatial dimension.

$$\frac{\partial I}{\partial t} = \frac{\dot{Z}x - \dot{X}\mu_s}{Z}\frac{\partial I}{\partial x} + \frac{\dot{Z}y - \dot{Y}\mu_s}{Z}\frac{\partial I}{\partial y} + \frac{\dot{Z}}{Z}\frac{c_E}{m}\left(\frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}\right) \tag{69}$$

Equations (9)-(11) can be applied to these coefficients as before, with (11) using v_0 for $\dot{X}$ and v_1 for $\dot{Y}$, for a complete solution of scene geometry. This is, again, in contrast with everywhere-in-focus images, which allow access to only v_0, v_1, and v_2.

C. Sensitivity - Depth from Sinusoidal Texture

We consider a sinusoidal texture of amplitude B_0, frequency f_0, and phase ϕ_0. A pinhole camera captures the in-focus image volume Q as below.

$$Q(x(t),t) = B_0 \sin(2\pi f_0[-Zx/\mu_s + X] + \phi_0) \tag{70}$$

We can express this equation in terms of image frequency $\omega(t)$ and phase $\phi(t)$.

$$Q(x(t),t) = B_0 \sin(\omega(t)x + \phi(t)), \tag{71}$$

$$\omega(t) = -2\pi f_0 Z/\mu_s, \tag{72}$$

$$\phi(t) = 2\pi f_0 X + \phi_0. \tag{73}$$

As before, we consider blurring the image by introducing an aperture with function $k(a)$ that induces an on-image blur kernel $h(x,t)$. We consider a narrow Gaussian function with standard deviation Σ. This induces an on-image Gaussian blur with standard deviation $\sigma(t)$, as below.

$$h(x,t) = \frac{\int_{-\infty}^{\infty} e^{-a^2/2\Sigma^2} da}{\int_{-\infty}^{\infty} e^{-x^2/2\sigma^2} dx} e^{-\frac{x^2}{2\sigma(t)^2}} \tag{74}$$

$$= \frac{\Sigma}{|\sigma(t)|} e^{-\frac{x^2}{2\sigma(t)^2}}, \tag{75}$$

$$\sigma(t) = \left|\frac{1}{Z(t)} - \frac{1}{\mu_f}\right| \mu_s \Sigma. \tag{76}$$

Because the Gaussian is a symmetric kernel, blurring with it changes only the contrast of the single sinusoid, which takes the value of the max-aligned integral. By the narrowness assumption, we ignore the subtractive correction to the Gaussian as before.

$$B(t) = \int \frac{\Sigma}{\sigma} e^{-\frac{x^2}{2\sigma^2}} B_0 \cos(\omega(t)x) dx \tag{77}$$

$$= B_0 \Sigma e^{-\frac{\sigma^2\omega^2}{2}} \tag{78}$$

Then, the blurred image volume is described by the equation below.

$$J(x(t),t) = B(t)\sin(\omega(t)x + \phi(t)) \tag{79}$$

The changing amplitude $B(t)$ of blurred image J encodes depth as follows. A good deal of arithmetic has been omitted, and the camera constant c_E defined in equation (46) is used.

$$\dot{B} = B_0 \Sigma(-\sigma\dot{\sigma}\omega^2 - \sigma^2\omega\dot{\omega}) e^{-\frac{\sigma^2\omega^2}{2}} \tag{80}$$

$$= (-\sigma\dot{\sigma}\omega^2 - \sigma^2\omega\dot{\omega})B, \tag{81}$$

$$-\frac{\dot{B}}{B\omega\dot{\omega}} = \frac{\sigma\dot{\sigma}\omega}{\dot{\omega}} + \sigma^2 \tag{82}$$

$$= c_E \frac{Z - \mu_f}{Z}, \tag{83}$$

$$Z = \frac{\mu_f c_E B\omega\dot{\omega}}{\dot{B} + c_E B\omega\dot{\omega}}. \tag{84}$$

Derivatives of the equation above lead to the propagation of error bound in the main text.