This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

LiFF: Light Field Features in Scale and Depth

Donald G. Dansereau[1,2], Bernd Girod[1], and Gordon Wetzstein[1]
[1]Stanford University, [2]The University of Sydney
donald.dansereau@sydney.edu.au

Abstract

Feature detectors and descriptors are key low-level vision tools that many higher-level tasks build on. Unfortunately these fail in the presence of challenging light transport effects including partial occlusion, low contrast, and reflective or refractive surfaces. Building on spatio-angular imaging modalities offered by emerging light field cameras, we introduce a new and computationally efficient 4D light field feature detector and descriptor: LiFF. LiFF is scale invariant and utilizes the full 4D light field to detect features that are robust to changes in perspective. This is particularly useful for structure from motion (SfM) and other tasks that match features across viewpoints of a scene. We demonstrate significantly improved 3D reconstructions via SfM when using LiFF instead of the leading 2D or 4D features, and show that LiFF runs an order of magnitude faster than the leading 4D approach. Finally, LiFF inherently estimates depth for each feature, opening a path for future research in light field-based SfM.

1. Introduction

Feature detection and matching are the basis for a broad range of tasks in computer vision. Image registration, pose estimation, 3D reconstruction, place recognition, combinations of these, e.g. structure from motion (SfM) and simultaneous localisation and mapping (SLAM), along with a vast body of related tasks, rely directly on being able to identify and match features across images. While these approaches work relatively robustly over a range of applications, some remain out of reach due to poor performance in challenging conditions. Even infrequent failures can be unacceptable, as in the case of autonomous driving.

State-of-the-art features fail in challenging conditions including self-similar, occlusion-rich, and non-Lambertian scenes, as well as in low-contrast scenarios including low light and scattering media. For example, the high rate of self-similarity and occlusion in the scene in Fig. 1 cause the COLMAP [35] SfM solution to fail. There is also an inherent tradeoff between computational burden and robustness:

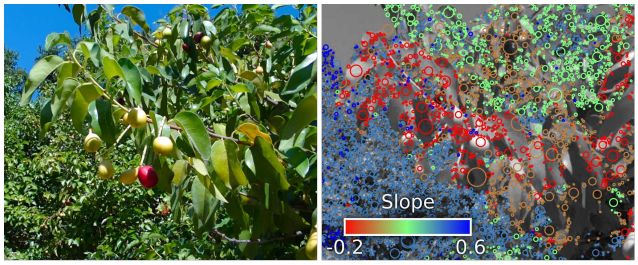

Figure 1. (left) One of five views of a scene that COLMAP's structure-from-motion (SfM) solution fails to reconstruct using SIFT, but successfully reconstructs using LiFF; (right) LiFF features have well-defined scale and depth, measured as light field slope, revealing the 3D structure of the scene – note we do not employ depth in the SfM solution. Code and dataset are at http://dgd.vision/Tools/LiFF, see the supplementary information for dataset details.

given sufficient computation it may be possible to make sense of an outlier-rich set of features, but it is more desirable to begin with higher-quality features, reducing computational burden, probability of failure, power consumption, and latency.

Light field (LF) imaging is an established tool in computer vision offering advantages in computational complexity and robustness to challenging scenarios [7,10,29,38,48]. This is due both to a more favourable signal-to-noise ratio (SNR) / depth of field tradeoff than for conventional cameras, and to the rich depth, occlusion, and native non-Lambertian surface capture inherently supported by LFs.

In this work we propose to detect and describe blobs directly from 4D LFs to deliver more informative features compared with the leading 2D and 4D alternatives. Just as the scale invariant feature transform (SIFT) detects blobs with well-defined scale, the proposed light field feature (LiFF) identifies blobs with both well-defined scale and well-defined depth in the scene. Structures that change their appearance with viewpoint, for example those refracted through or reflected off curved surfaces, and those formed by occluding edges, will not satisfy these criteria. At the same time, well-defined features that are partially occluded are not normally detected by 2D methods, but can be detected by LiFF via focusing around partial occluders.

Ultimately LiFF features result in fewer misregistrations, more robust behaviour, and more complete 3D models than the leading 2D and 4D methods, allowing operation over a broader range of conditions. Following recent work comparing hand-crafted and learned features [36], we evaluate LiFF in terms of both low-level detections and the higher-level task of 3D point cloud reconstruction via SfM.

LiFF features have applicability where challenging conditions arise, including autonomous driving, delivery drones, surveillance, and infrastructure monitoring, in which weather and low light commonly complicate vision. It also opens a range of applications in which feature-based methods are not presently employed due to their poor rate of success, including medical imagery, industrial sites with poor visibility such as mines, and in underwater systems.

The key contributions of this work are:

- We describe LiFF, a novel feature detector and descriptor that is less computationally expensive than leading 4D methods and natively delivers depth information;
- We demonstrate that LiFF yields superior detection rates compared with competing 2D and 4D methods in low-SNR scenarios; and
- We show that LiFF extends the range of conditions under which SfM can work reliably, outperforming SIFT in reconstruction performance.

To evaluate LiFF we collected a large multi-view LF dataset containing over 4000 LFs of over 800 scenes. This is the first large dataset of its kind, with previous examples limited to a single LF of each scene [39]. It is our hope that LiFF and the accompanying dataset will stimulate a broad range of research in feature detection, registration, interpolation, SfM, and SLAM.

2. Related Work

Feature Detection and Matching 2D feature detectors such as SIFT [25], SURF [2], FAST [33], and ORB [34], are instrumental in many computer vision algorithms, including SfM, SLAM, disparity estimation, and tracking. Many of these applications rely on matching features between different viewpoints of the same scene. Unfortunately, this matching is often unreliable, because similar spatial structures can occur several times in the same scene, and view-dependent effects such as partial occlusion and specularity makes features look different from different perspectives. To reliably match features, additional geometric constraints have to be imposed, for example via bundle adjustment, but this is computationally expensive, severely affecting runtime, required memory, and power.

3D feature detection from RGB-D images can be more robust than 2D feature detection, as demonstrated in the context of object detection and segmentation [17] as well as SLAM [12]. Rather than working with RGB-D data, 3D feature detectors can also operate directly on point clouds [16, 40, 55] while providing similar benefits. However, point clouds are usually not available in conventional imaging systems and RGB-D data does not generally handle partial occlusion and other view-dependent effects.

LFs inherently capture a structured 4D representation that includes view-dependent effects, partial occlusions, and depth. A number of existing works touch upon exploiting these characteristics for feature detection and description. Ghasemi et al. [14] exploit depth information in the LF to build a global, scale-invariant descriptor useful for scene classification, though they do not address the localized features required for 3D reconstruction. Tosic et al. [44] employ LF scale and depth to derive an edge-sensitive feature detector. Our focus is on blob detection rather than edge detection since it is much easier to uniquely match blobs across viewpoints, making it appropriate for a larger set of tasks including 3D reconstruction.

The leading method for extracting features from LFs is to run a 2D detector across subimages, then consolidate the detected features by imposing consistency with epipolar geometry. For example, Teixeira et al. [42] propose feature detection that repeats SIFT on 2D subimages then consolidates across 2D epipolar slices. In exploring LF SfM, Johannsen et al. [20] extract SIFT features across subimages then consolidate them using 4D LF geometry. Zhang et al. [54] demonstrate that line and plane correspondence can be employed for LF-based SfM, by detecting 2D line segments in subimages then applying a higher-order consolidation step. Finally, Maeno et al. [26] and Xu et al. [53] detect refractive objects by following 2D features through the LF using optical flow. They then enforce 4D epipolar geometry to detect refracted features.

While these approaches differ in their details they are all fundamentally limited by the performance of the 2D detector they build upon. We refer to these as repeated 2D detectors, and make direct comparison to repeated SIFT in this work. We show that LiFF shows significantly higher performance over repeated 2D methods by virtue of simultaneously considering all subimages in detecting and describing features. Because repeated 2D detectors are less direct in their approach, they present more parameters requiring tuning, making them more difficult to deploy. Finally, repeating SIFT across viewpoints is a highly redundant operation, and we will show that LiFF has significantly lower computational complexity.

Light Field Imaging An LF [15, 23] contains 4D spatio-angular information about the light in a scene, and can be recorded with a camera array [51], or a sensor equipped with a lenslet array [1, 8, 31] or a coded mask [28, 45]. See [19, 50] for detailed overviews of LF imaging. To date, LF image processing has been applied to a variety of applications including image-based rendering [9, 22, 23],

post-capture image refocus [13, 30], SfM [20], lens aberration correction [18], spatial [3] and temporal [46] super-resolution, video stabilization [37], motion deblurring [38], and depth imaging [24, 41, 43, 47, 49]. In this work, we explore robust LF feature detection and matching for improving applications in reconstruction including SfM.

Conventions In this work we consider the two-plane-parameterized LF $L(s,t,u,v)$ with $N_s \times N_t$ views of $N_u \times N_v$ pixels each [7, 23]. A point in 3D space appears in the LF as a plane with slope inversely proportional to the point's depth [1, 5, 19]. Working with sampled LFs introduces unknown scaling factors between slope and depth, which can either be tolerated or calibrated away. In the following we refer to slope with the understanding that it can be mapped to depth via camera calibration [4, 6, 52].

3. Light Field Feature Detection

We begin our development with the well-known SIFT feature detector and extend it to 4D LFs. We begin with SIFT because of its dominance in reconstruction applications [36]. Our key insight is that while SIFT locates blobs with well-defined scales and locations in the 2D image plane, LFs offer the ability to identify blobs with well-defined scales and locations *in 3D space*.

To generalize SIFT to the LF we first propose a much more computationally expensive approach that searches for features in a joint 4D scale-slope space. We then show how numerically identical results can be achieved by first converting the LF to a focal stack while retaining the 4D search step. The result is both more robust and more computationally efficient than repeating SIFT across the LF. This approach offers numerous advantages including rejection of undesired spurious features at occlusion boundaries, detection of desired but partially occluded features, and inherent depth estimation.

SIFT identifies blobs by searching for extrema in a 3D scale space constructed as a difference of Gaussian (DoG) stack. The DoG is built by convolving with a set of Gaussian filters covering a range of scales, then taking the difference between adjacent scales, as in

$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y), \tag{1}$$
$$D(x,y,\sigma_i) = L(x,y,\sigma_{i+1}) - L(x,y,\sigma_i), \tag{2}$$

where $G(x,y,\sigma)$ is a Gaussian filter at scale σ, and the DoG is computed over a range of scales $\sigma_i, 1 \leq i \leq N$ with constant multiplicative factor k such that $\sigma_{i+1} = k\sigma_i$.

The convolutions (1) represent the bulk of the computational cost of SIFT. Significant savings can be had by applying larger-scaled convolutions on downsampled versions of the input image [25]. Nevertheless, a good approximation of the cost of this approach is to understand it as a set of N 2D filtering operations, which we denote $N \times \text{Filt}_{\text{2D}}$.

Following extrema detection, SIFT proceeds through steps for sub-pixel-accurate feature location, rejection of edge features that can trigger the blob detection process, and estimation of dominant orientation allowing rotation invariance. Finally, an image descriptor is constructed from histograms of edge orientations. LiFF will differ from these steps only in the detection and descriptor stages.

3.1. Searching Scale and Slope

Jointly searching across scale and 3D position can be accomplished as a direct extension of SIFT's DoG space. We first rewrite each scale of the DoG (2) as a single convolution, applied in the u and v dimensions

$$H_\sigma(u,v,\sigma) = G(u,v,\sigma_{i+1}) - G(u,v,\sigma), \tag{3}$$
$$D_{2D}(u,v,\sigma) = H_\sigma(u,v,\sigma) * I(u,v). \tag{4}$$

The filter H_σ finds blobs in LF subimages at the scale σ. We augment this with depth selectivity using a frequency-planar filter H_λ. The frequency-planar filter selects for a specific depth in the LF, and can be constructed in a number of ways in the frequency or spatial domains [7, 30]. For this work we consider the direct spatial-domain implementation

$$H_\lambda(s,t,u,v,\lambda) = \begin{cases} 1, & u = \lambda s,\ v = \lambda t, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

We combine (3) and (5) to yield a filter that is simultaneously selective in scale and slope:

$$H(\boldsymbol{\phi},\sigma,\lambda) = H_\sigma(u,v,\sigma) * H_\lambda(\boldsymbol{\phi},\lambda), \tag{6}$$

where $\boldsymbol{\phi} = [s,t,u,v]$ gathers the LF indices. We apply the filter H over N scales σ and M slopes λ:

$$D_{6D}(\boldsymbol{\phi},\sigma,\lambda) = H(\boldsymbol{\phi},\sigma,\lambda) * L(\boldsymbol{\phi}). \tag{7}$$

D_{6D} is highly redundant in that each subimage contains virtually the same information, and so when searching for local extrema we restrict our attention to the central view in s,t yielding the 4D search space $D(u,v,\sigma,\lambda)$.

Identifying local extrema in D is a straightforward extension of the 3D approach used in SIFT, yielding feature coordinates $[u,v,\sigma,\lambda]$. It is important to jointly search the scale-slope space in order to identify those features with both distinct scale and slope. This is a key distinction between LiFF and repeating SIFT over the LF or a focal stack.

3.2. Simplification using Focal Stack

The method so far is extremely computationally expensive. The 4D convolution (7) is repeated over N scales and M slopes. The key insight in simplifying (7) is exploiting the linear separability of H_σ and H_λ seen in (6). The fact that we employ only the central view of D allows the slope

selectivity step to be computed only over that subset, collapsing the 4D LF into a 3D focal stack:

$$F(u, v, \lambda) = \sum_{s,t} L(s, t, u - \lambda s, v - \lambda t), \tag{8}$$

$$D(u, v, \sigma, \lambda) = H_\sigma(u, v, \sigma) * F(u, v, \lambda). \tag{9}$$

i.e. we compute a focal stack F over M slopes, then apply a DoG filter over N scales for each slope. Finally, we search the joint space D for extrema. This process yields numerically identical results to building the full 6D scale-slope space (7), but at a fraction of the computational cost.

A few efficient methods for computing the focal stack F have been proposed [27, 32]. These generally find at minimum as many layers as there are samples in s or t. Feature detection may not require so many layers, and so we proceed with the more straightforward approach of shifting and summing LF subimages (8), with the understanding that computational savings may be possible for large stack depths. The cost of this focal stack is $M \times N_s \times N_t \times N_u \times N_v$.

Computing the DoG from each focal stack image F is identical to the first steps of conventional SIFT, and can benefit from the same downsampling optimization [25]. We approximate the complexity as M times the cost of conventional SIFT, $M \times N \times \mathrm{Filt}_{2\mathrm{D}}$. For practical scenarios this will overshadow the cost of computing the focal stack.

3.3. Feature Descriptor

As with SIFT, for each feature $[u, v, \sigma, \lambda]$ we construct a histogram of edge orientations. The key difference with the LiFF descriptor is that it is computed at a specific depth in the scene corresponding to the detected slope λ. Each descriptor is thus constructed from the appropriate stack slice $F(u, v, \lambda)$. The key advantage is selectivity against interfering objects at different depths including partial occluders and reflections off glossy surfaces.

3.4. Complexity

A common approach to LF feature detection is repeating SIFT across subimages, then applying a consistency check to reject spurious detections [20,42]. The complexity of this approach is at least the cost of the DoG operations applied over the subimages, i.e. $N_s \times N_t \times N \times \mathrm{Filt}_{2\mathrm{D}}$. Note that this ignores the cost of consolidating observations across views, which varies by implementation and can be substantial.

Comparing complexity, we see that for M slopes LiFF is at least $N_s N_t / M$ times faster than repeated SIFT. In a typical scenario using Lytro Illum-captured LFs with 11×11 views, and applying LiFF over $M = 11$ slopes, LiFF will be about 11 times faster than repeated SIFT. For larger LFs, e.g. Stanford gantry-collected LFs[1] with 17×17 views, the speed increase is larger, 26 times, assuming the same slope count. When accounting for time required to consolidate features across views in repeated SIFT, the speed gain is even larger.

[1] http://lightfields.stanford.edu

3.5. Parameters

LiFF has the same parameters as SIFT: a list of scales at which to compute the DoG, a peak detection threshold, and an edge rejection threshold. The descriptor parameters are also the same, including the area over which to collect edge histograms, numbers of bins, and so on. The only additional parameter for LiFF is a list of slopes over which to compute the focal stack. A good rule of thumb for lenslet-based cameras is to consider slopes between -1 and 1, with as many slopes as there are samples in N_s or N_t. Larger slope counts increase compute time without improving performance, while smaller slope counts can miss features at specific depths in the scene.

4. Evaluation

LiFF Implementation Our implementation of LiFF is in C, compiled into MEX files that we call from MATLAB. For testing purposes, we load light fields and convert to grayscale in MATLAB, but the feature detection and extraction process is entirely in C. Our focal stack implementation uses the shift-and-sum method with nearest-neighbour interpolation, and includes a normalization step which prevents darkening near the edges of the LF.

Repeated SIFT Implementation To compare LiFF with repeated SIFT, we called the VLFeat C implementation of SIFT v0.9.21, and in MATLAB implemented a consolidation process that enforces consistency between subimages. A variety of approaches have been suggested [20,26,42,53]. Our goal for SfM testing is not speed, but approaching the upper bound of performance. We therefore employ an exhaustive search starting at each detected 2D feature across all subimages. For each feature we identify matching detections in all other subimages based on a set of criteria including scale, orientation, feature descriptor, and maximum deviation from a best-fit plane. When evaluating speed we omit the time taken for this consolidation process.

A key parameter of any repeated 2D detector is the number of subimages in which a feature must be identified before being considered a detection. In the following we test across different thresholds, and identify the method accordingly, e.g. repeated SIFT 0.5 requires that at least half of the subimages contain a detected feature.

Our repeated SIFT implementation is not very computationally efficient. However we believe its performance is revealing of a broad class of repeated and consolidated 2D features.

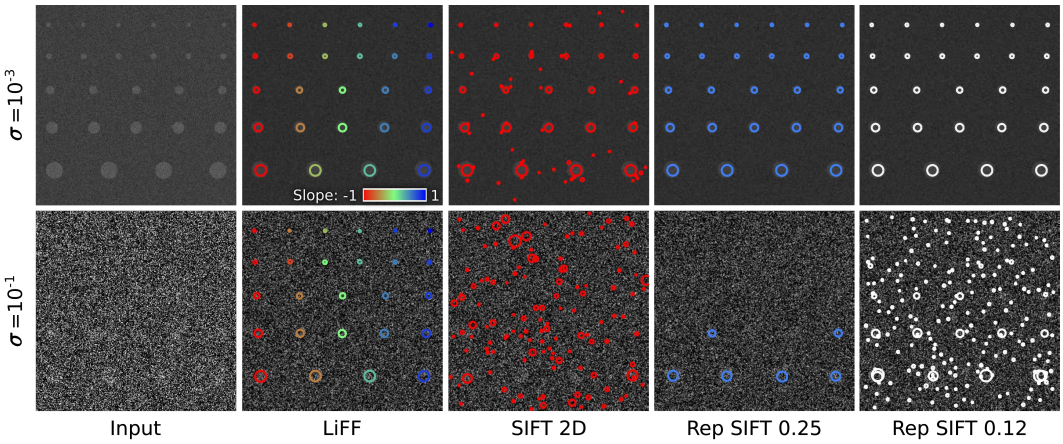

Figure 2. A set of disks at varying scales and depths, presented at two noise levels σ. At the lower noise level (top row), all methods operate reasonably well; while SIFT shows some spurious detections, repeated SIFT is able to reject these by imposing consistency between views; In higher noise (bottom row) LiFF's performance is ideal including reasonable slope estimates, but SIFT misses some features and has spurious detections; repeated SIFT with threshold 0.25 rejects the spurious features but cannot locate those missed in the individual views; and repeated SIFT with a lower threshold admits more spurious detections while still missing some true positives.

4.1. Speed

We compared the speed of our LiFF implementation with the SIFT implementation in VLFeat. All tests were run on an Intel i7-8700 at 3.20 GHz. The test included both feature detection and descriptor extraction, and was run on scenes with similar feature counts for SIFT and LiFF. On Illum-captured LFs with $11 \times 11 \times 541 \times 376$ samples, we found LiFF took on average 2.88 sec, while repeating SIFT across subimages took on average 53.1 sec, excluding time to consolidate observations, which was considerable.

Overall the speed increase moving from repeated SIFT to LiFF with our implementation is measured as $18\times$, which agrees well with the anticipated speed gain. Further speed improvements should be possible: as with SIFT, LiFF is amenable to optimization via parallelization, implementation on GPU, etc.

4.2. Noise Performance

Repeated SIFT is fundamentally constrained by the performance of the 2D method it builds on. To demonstrate this we synthesized a set of scenes with known good feature locations, and introduced varying levels of noise to observe feature performance.

In one set of experiments, depicted in Fig. 2, the input consists of 26 disks at varying scales and at depths corresponding to slopes between -1 and 1. The LF has dimensions $9 \times 9 \times 256 \times 256$ and a signal contrast of 0.1. We introduced moderate noise with variance 10^{-3} (top), and strong noise with variance 10^{-1} (bottom).

We ran SIFT operating on the central subimage of the LF, repeated SIFT with minimum subimage agreements of 0.25 and 0.12, and LiFF. The common parameters of peak threshold, edge detection threshold, and scale range were identical for all methods. LiFF was operated over 9 slopes between -1 and 1.

As seen in Fig. 2, LiFF successfully detects all 26 disks in both moderate and high noise, as well as providing slope estimates even in high noise. SIFT suffers from spurious

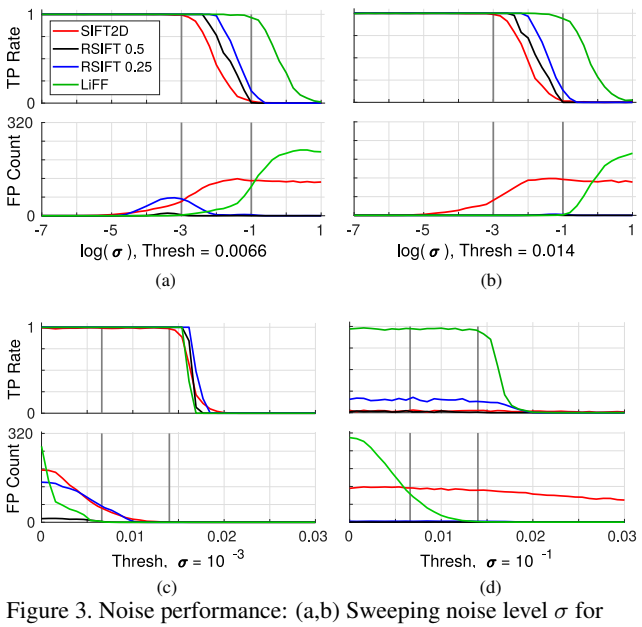

Figure 3. Noise performance: (a,b) Sweeping noise level σ for fixed detection thresholds, LiFF has the best true positive (TP) rate for noisy imagery, though like SIFT suffers from a high false positive (FP) count; (c) Sweeping detection threshold, the methods show similar performance in moderate noise, while (d) LiFF delivers a much higher TP rate and zero FP rate in high noise for appropriately set threshold. Overall, LiFF matches or outperforms both SIFT and repeated SIFT.

Method	% pass	Keypts / Img	Putative Matches / Img	Inlier Matches / Img	Match Ratio	Precision	Matching Score	3D Points	Track Len
COLMAP Defaults									
LiFF	**64.19**	2684	**282**	**274**	**0.14**	**0.96**	**0.13**	**382**	**3.38**
SIFT	57.83	2669	243	235	0.10	0.95	0.10	337	3.31
COLMAP Permissive									
LiFF	97.53	2689	**213**	**206**	**0.11**	**0.93**	**0.11**	**472**	**2.46**
SIFT	**97.88**	2688	175	167	0.077	0.92	0.073	396	2.40
Defaults Intersect									
LiFF	54.65	2674	**304**	**297**	**0.15**	**0.96**	**0.14**	**418**	**3.44**
SIFT	54.65	2689	248	240	0.10	0.95	0.10	348	3.33
Permissive Intersect									
LiFF	96.23	2687	**212**	**205**	**0.11**	**0.93**	**0.11**	**473**	**2.46**
SIFT	96.23	2684	172	165	0.076	0.92	0.073	397	2.40

Table 1. Structure-from-motion: With COLMAP's default values, LiFF outperforms SIFT in all measures, including successful reconstruction of significantly more scenes; with more permissive settings, COLMAP reconstructs nearly all scenes, succeeding on slightly more scenes using SIFT, but with LiFF outperforming SIFT in all other measures including 3D points per model. Taking only those scenes that passed with both feature detectors ("Intersect") allows a direct comparison of performance, with LiFF outperforming SIFT in all cases.

detections in moderate noise, and both missed and spurious detections in high noise. Repeated SIFT successfully rejects spurious detections in low noise, but either misses detections or both misses detections and admits spurious features in high noise, depending on its threshold.

To better expose the behaviours of these methods we ran a set of experiments on the same scene with varying noise levels and peak detection thresholds, measuring true positive (TP) rate over the 26 disks, and false positive (FP) count. Each experiment was repeated 25 times, with the mean results shown in Fig. 3. The top row depicts two detection thresholds (highlighted as vertical bars on the bottom row), with noise variances σ swept between 10^{-7} and 10^{1}. The TP rate shows that LiFF correctly detects features in more than an order of magnitude higher noise than the other methods. At high noise levels LiFF and SIFT both suffer from high FP counts, though this is somewhat ameliorated for LiFF by setting a higher peak detection threshold.

The bottom row of Fig. 3 depicts two noise levels, $\sigma = 10^{-3}$ and 10^{-1} (highlighted as vertical bars in the top row), for varying peak detection thresholds. In moderate noise (left) all methods perform similarly across a range of threshold values. In high noise (right), only LiFF delivers a good TP rate, and a nil FP count for a sufficiently large detection threshold.

From these experiments we conclude that LiFF offers enhanced performance in noisy conditions compared with SIFT and repeated SIFT. We expect this increased performance applies to LFs collected in low light, and also to shadowed and low-contrast regions of well-lit scenes. It also applies where contrast is limited by participating media like water, dust, smoke, or fog.

4.3. Structure from Motion

Following the feature comparison approach in [36], we employed an SfM solution to evaluate LiFF in the context of 3D reconstruction applications. We used a Lytro Illum to collect a large dataset of LFs with multiple views of each scene. The dataset contains 4211 LFs covering 850 scenes in 30 categories, with between 3 and 5 views of each scene. Images are in indoor and outdoor campus environments, and include examples of Lambertian and non-Lambertian surfaces, occlusion, specularity, subsurface scattering, fine detail, and transparency. No attempt was made to emphasize challenging content.

Although we expect LiFF's slope estimates could dramatically improve SfM, we ignore this information to allow a more direct comparison with SIFT. We also use identical settings for all parameters common to SIFT and LiFF. Based on the noise performance experiments above, a higher peak threshold for LiFF would likely result in fewer spurious features without loss of useful features. However, by using identical thresholds we are better able to highlight the behavioural differences between LiFF and SIFT, rather than focusing exclusively on the difference in noise performance.

We extracted the central view of each LF and converted to grayscale. The method of grayscale conversion significantly impacts performance [21], and we determined that MATLAB's luminance-based conversion followed by gamma correction with a factor of 0.5, and finally histogram equalization, yielded good results.

We ran LiFF and the VLFeat implementation of SIFT using a peak threshold of 0.0066, edge threshold 10, and DoG scales covering 4 octaves over 3 levels per octave. We started at octave -1 because our images are relatively small,

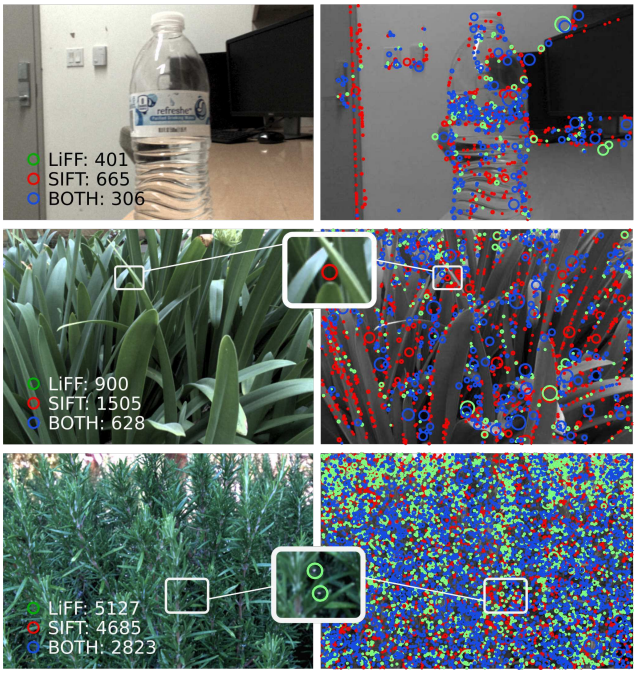

Figure 4. Comparison to SIFT: Features identified only by LiFF, only by SIFT, and by both are shown in green, red, and blue respectively. (top) LiFF rejects spurious features in low-contrast areas and to some extent those distorted through refraction; (center) LiFF rejects spurious features at occlusion boundaries – the inset highlights a SIFT-only detection caused by leaves at different depths; (bottom) LiFF detects partially occluded features missed by SIFT – note the increasing proportion of LiFF-only features toward the back of the scene, and the LiFF-only detections highlighted in the inset. Slope estimates for the bottom scene are shown in Fig. 5.

making smaller features important. For LiFF we employed the centermost 11×11 subimages, and computed the focal stack over 11 slopes between -1 and 1.

For the feature descriptor we found that L1 root normalization yields significantly improved matching compared with the default L2 normalization build into VLFeat's implementation of SIFT. We therefore applied this same normalization scheme to both SIFT and LiFF feature descriptors. To confirm that our external feature detection was working correctly, we compared COLMAP's performance when using our externally extracted SIFT features and when using its internal calls to SIFT, and achieved virtually identical results.

We ran COLMAP up to and including the SfM stage, stopping before dense multi-view stereo reconstruction. We evaluated performance in terms of numbers of keypoints per image, putative feature matches generated per image, and number of putative matches classified as inliers during SfM. Following [36], we also evaluated the putative match ratio: the proportion of detected features that yield putative matches; precision: the proportion of putative matches yielding inlier matches; matching score: the proportion of features yielding inlier matches; the mean number of 3D points in the reconstructed models; and track length: the mean number of images over which a feature is successfully tracked.

With its default settings, we found that COLMAP failed to generate output for many scenes. It failed to converge during bundle adjustment, or failed to identify a good initial image pair. With each of our images having only 541×376 pixels, and each scene only 3 to 5 images, COLMAP's default settings are not well suited to our dataset. The difference in performance between LiFF and SIFT at this stage is nevertheless informative, and is shown in the top row of Table 1. LiFF did not detect many more features than SIFT, but it did result in a significantly higher number of successfully reconstructed scenes (% pass). The statistics support the conclusion that LiFF has a higher proportion of informative features, yielding higher absolute numbers of putative and inlier matches, higher proportions of inlier matches, more 3D points, and longer track lengths. Note that we have not highlighted the higher keypoint count as being a superior result, as having LiFF detect more features is not necessarily a better outcome without those features also being useful.

We relaxed COLMAP's settings to better deal with our dataset, reducing the mapper's minimum inlier counts, minimum track length, and minimum 3D point count. In this more permissive mode COLMAP was able to reconstruct most of the scenes in the dataset. As seen in the second set of results in Table. 1, in this mode SIFT allowed slightly more scenes to be reconstructed, and detected a nearly identical number of features, but performed dramatically less well than LiFF in all other statistics. Note in particular that LiFF-generated models had on average 472 reconstructed points compared with SIFT's 396.

A shortcoming of the comparisons made above is that they are applied over different subsets of the data: SIFT passed a different set of scenes than LiFF. For a fair comparison we computed the same statistics over only those scenes that passed using both SIFT and LiFF features. The results, in the bottom half of Table 1, clearly show LiFF outperforming SIFT in all measures.

4.4. Challenging Cases

To better expose the differences in performance between SIFT and LiFF, we investigated those scenes for which COLMAP had trouble converging with SIFT features, but passed when using LiFF features. Fig. 4 depicts some informative examples. At right we show features detected only by LiFF (green), only by SIFT (red), and by both methods (blue). In the top row we see that this relatively well-lit indoor scene has low contrast around the door edge yielding many spurious SIFT-only detections. Note also that the tex-

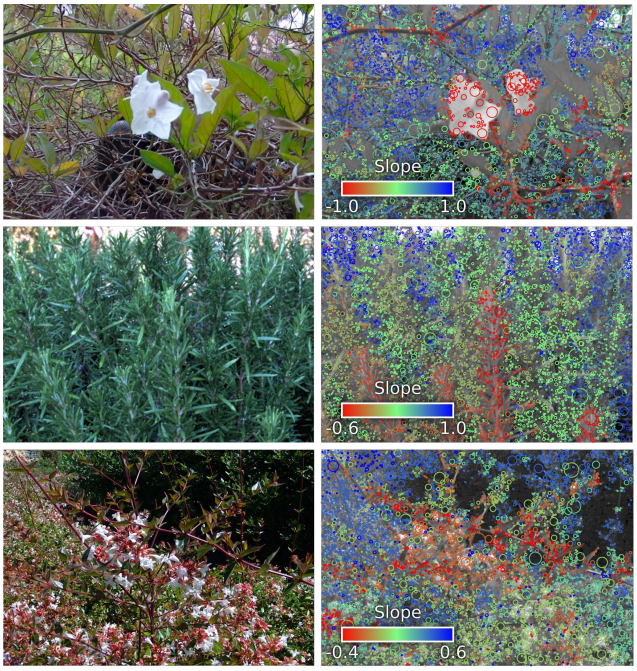

Figure 5. 3D Scene Shape: In this work we establish LiFF's ability to deliver more informative features by virtue of higher selectivity and ability to image through partial occlusion. We expect LiFF's slope estimates will also be of substantial interest. Here we see the 3D shape of each scene revealed through the slopes of the detected LiFF features.

ture refracted through the water bottle triggers some SIFT-only detections. The inconsistent apparent motion of refracted features make them undesirable for SfM, and the lack of a well-defined depth prevents LiFF from detecting these as features.

The center row in Fig. 4 shows a scene with many spurious SIFT detections near edges, but also at occlusion boundaries. SIFT cannot distinguish between well-defined shapes and those formed by the chance alignment of occluding objects. LIFF on the other hand rejects shapes formed by occluding objects at different depths, as these do not have a well-defined depth. A typical spurious occlusion feature detected only by SIFT is highlighted in the inset.

The bottom row in Fig. 4 shows a scene for which LiFF delivers more features than SIFT. Notice the increasing proportion of LiFF-only features towards the back of the scene, where most of the features are partially occluded by foreground elements. In the inset we see an example of two water droplets just visible through foreground occlusions, detected only by LiFF. In more extreme cases, features may be entirely blocked in some subimages but still visible to LiFF. Note that the green circles in the inset are expanded to aid clarity. This scene is repeated in Fig. 5, which provides visual confirmation that 3D structure is being reflected in the LiFF slope estimates.

5. Conclusion

We presented LiFF, a feature detector and descriptor for LFs that directly extends SIFT to operate on the entire LF. The proposed detector is faster than the common practice of repeating SIFT across multiple views, and produces more correct detections and fewer spurious detections in challenging conditions. We demonstrate an $18\times$ speed increase on Lytro Illum-captured imagery compared with repeated SIFT, and anticipate further optimization is possible via parallelization and implementation on GPU.

In SfM tests, we showed LiFF to outperform SIFT in terms of absolute numbers of putative and inlier matches, proportions of inlier matches, numbers of images through which features are tracked, and the numbers of 3D points in the reconstructed models. Our test dataset was not manipulated to emphasize challenging scenes, these results are for typical indoor and outdoor environments. We expect that in more challenging conditions LiFF can even more dramatically improve the performance of 3D reconstruction, and expand the range of applications in which feature-based techniques can be applied.

As future work we expect that adaptive selection of focal stack slopes could further improve the speed of LiFF. An interesting benefit of the focal stack is that it can be trivially extended to perform linear super-resolution [19], allowing finer features to be detected, though at the cost of increased processing time. An exploration of the application of LiFF to directly captured focal stacks might also prove interesting.

Recent work has shown that computing histograms over multiple scales offers improved SIFT detector performance, and this can also be applied to LiFF features [11, 36]. We also anticipate the slope information that LiFF recovers to be of interest. For a calibrated LF camera, slope yields an absolute 3D position and absolute scale for each feature. This absolute scale can be employed as a discriminator in a scale-*sensitive* approach to feature matching. Finally, the 3D information retrieved by LiFF may be of significant utility in directly informing 3D reconstruction.

Acknowledgments This work was supported in part by the NSF/Intel Partnership on Visual and Experiential Computing (Intel #1539120, NSF #IIS-1539120).

References

[1] E. H. Adelson and J. Y. A. Wang. Single lens stereo with a plenoptic camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 14(2):99–106, 1992. 2, 3

[2] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. Speeded-up robust features (SURF). *Computer Vision and image understanding*, 110(3):346–359, 2008. 2

[3] T. E. Bishop, S. Zanetti, and P. Favaro. Light field superresolution. In *Computational Photography (ICCP)*, pages 1–9. IEEE, 2009. 3

[4] Y. Bok, H.-G. Jeon, and I. S. Kweon. Geometric calibration of micro-lens-based light-field cameras using line features. In *Computer Vision–ECCV 2014*, pages 47–61. Springer, 2014. 3

[5] R. Bolles, H. Baker, and D. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *Intl. Journal of Computer Vision (IJCV)*, 1(1):7–55, 1987. 3

[6] D. G. Dansereau, O. Pizarro, and S. B. Williams. Decoding, calibration and rectification for lenselet-based plenoptic cameras. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1027–1034. IEEE, June 2013. 3

[7] D. G. Dansereau, O. Pizarro, and S. B. Williams. Linear volumetric focus for light field cameras. *ACM Transactions on Graphics (TOG)*, 34(2):15, Feb. 2015. 1, 3

[8] D. G. Dansereau, G. Schuster, J. Ford, and G. Wetzstein. A wide-field-of-view monocentric light field camera. In *Computer Vision and Pattern Recognition (CVPR)*, pages 3757–3766. IEEE, July 2017. 2

[9] A. Davis, M. Levoy, and F. Durand. Unstructured light fields. In *Computer Graphics Forum*, volume 31, pages 305–314. Wiley Online Library, 2012. 2

[10] F. Dong, S.-H. Ieng, X. Savatier, R. Etienne-Cummings, and R. Benosman. Plenoptic cameras in real-time robotics. *Intl. Journal of Robotics Research (IJRR)*, 32(2):206–217, 2013. 1

[11] J. Dong and S. Soatto. Domain-size pooling in local descriptors: Dsp-sift. In *Computer Vision and Pattern Recognition (CVPR)*, pages 5097–5106, 2015. 8

[12] X. Gao and T. Zhang. Robust RGB-D simultaneous localization and mapping using planar point features. *Robotics and Autonomous Systems*, 72:1–14, 2015. 2

[13] T. Georgiev, C. Intwala, S. Babakan, and A. Lumsdaine. Unified frequency domain analysis of lightfield cameras. In *Computer Vision–ECCV 2008*, pages 224–237. Springer, 2008. 3

[14] A. Ghasemi and M. Vetterli. Scale-invariant representation of light field images for object recognition and tracking. In *Proceedings of the SPIE*, volume 9020. Intl. Society for Optics and Photonics, 2014. 2

[15] S. Gortler, R. Grzeszczuk, R. Szeliski, and M. Cohen. The lumigraph. In *SIGGRAPH*, pages 43–54. ACM, 1996. 2

[16] S. Gumhold, X. Wang, and R. S. MacLeod. Feature extraction from point clouds. In *IMR*, 2001. 2

[17] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik. Learning rich features from RGB-D images for object detection and segmentation: Supplementary material, 2014. 2

[18] P. Hanrahan and R. Ng. Digital correction of lens aberrations in light field photography. In *Intl. Optical Design Conference*, page WB2. Optical Society of America, 2006. 3

[19] I. Ihrke, J. Restrepo, and L. Mignard-Debise. Principles of light field imaging. *IEEE Signal Processing Magazine*, 1053(5888/16), 2016. 2, 3, 8

[20] O. Johannsen, A. Sulc, and B. Goldluecke. On linear structure from motion for light field cameras. In *Intl. Conference on Computer Vision (ICCV)*, pages 720–728, 2015. 2, 3, 4

[21] C. Kanan and G. W. Cottrell. Color-to-grayscale: does the method matter in image recognition? *PloS one*, 7(1):e29740, 2012. 6

[22] A. Levin and F. Durand. Linear view synthesis using a dimensionality gap light field prior. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1831–1838. IEEE, 2010. 2

[23] M. Levoy and P. Hanrahan. Light field rendering. In *SIGGRAPH*, pages 31–42. ACM, 1996. 2, 3

[24] C.-K. Liang and R. Ramamoorthi. A light transport framework for lenslet light field cameras. *ACM Transactions on Graphics (TOG)*, 34(2):16, 2015. 3

[25] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Intl. Journal of Computer Vision (IJCV)*, 60(2):91–110, 2004. 2, 3, 4

[26] K. Maeno, H. Nagahara, A. Shimada, and R. I. Taniguchi. Light field distortion feature for transparent object recognition. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2786–2793. IEEE, June 2013. 2, 4

[27] J. G. Marichal-Hernández, J. P. Lüke, F. L. Rosa, and J. M. Rodríguez-Ramos. Fast approximate 4D: 3D discrete radon transform, from light field to focal stack with o(n4) sums. In *IS&T/SPIE Electronic Imaging*, pages 78710G–78710G. Intl. Society for Optics and Photonics, 2011. 4

[28] K. Marwah, G. Wetzstein, Y. Bando, and R. Raskar. Compressive light field photography using overcomplete dictionaries and optimized projections. In *SIGGRAPH*, volume 32, pages 1–11, New York, NY, USA, 2013. ACM. 2

[29] K. Mitra, O. S. Cossairt, and A. Veeraraghavan. A framework for analysis of computational imaging systems: role of signal prior, sensor noise and multiplexing. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(10):1909–1921, 2014. 1

[30] R. Ng. Fourier slice photography. *ACM Transactions on Graphics (TOG)*, 24(3):735–744, July 2005. 3

[31] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan. Light field photography with a hand-held plenoptic camera. Technical report, Stanford University Computer Science, 2005. 2

[32] F. Pérez, A. Pérez, M. Rodríguez, and E. Magdaleno. A fast and memory-efficient discrete focal stack transform for plenoptic sensors. *Digital Signal Processing*, 2014. 4

[33] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *European Conference on Computer Vision (ECCV)*, pages 430–443. Springer, 2006. 2

[34] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: An efficient alternative to SIFT or SURF. In *Intl. Conference on Computer Vision (ICCV)*, pages 2564–2571. IEEE, 2011. 2

[35] J. L. Schonberger and J.-M. Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016. 1

[36] J. L. Schönberger, H. Hardmeier, T. Sattler, and M. Pollefeys. Comparative evaluation of hand-crafted and learned local features. In *Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2, 3, 6, 7, 8

[37] B. Smith, L. Zhang, H. Jin, and A. Agarwala. Light field video stabilization. In *Intl. Conference on Computer Vision (ICCV)*, pages 341–348. IEEE, 2009. 3

[38] P. P. Srinivasan, R. Ng, and R. Ramamoorthi. Light field blind motion deblurring. In *Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 3

[39] P. P. Srinivasan, T. Wang, A. Sreelal, R. Ramamoorthi, and R. Ng. Learning to synthesize a 4D RGBD light field from a single image. In *Intl. Conference on Computer Vision (ICCV)*, pages 2262–2270, 2017. 2

[40] B. Steder, R. B. Rusu, K. Konolige, and W. Burgard. NARF: 3D range image features for object recognition. In *Intelligent Robots and Systems (IROS) Workshops*, volume 44, 2010. 2

[41] M. W. Tao, P. P. Srinivasan, J. Malik, S. Rusinkiewicz, and R. Ramamoorthi. Depth from shading, defocus, and correspondence using light-field angular coherence. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1940–1948, 2015. 3

[42] J. A. Teixeira, C. Brites, F. Pereira, and J. Ascenso. Epipolar based light field key-location detector. In *Multimedia Signal Processing (MMSP)*, pages 1–6. IEEE, 2017. 2, 4

[43] J. Tian, Z. Murez, T. Cui, Z. Zhang, D. Kriegman, and R. Ramamoorthi. Depth and image restoration from light field in a scattering medium. In *Intl. Conference on Computer Vision (ICCV)*, 2017. 3

[44] I. Tošić and K. Berkner. 3D keypoint detection by light field scale-depth space analysis. In *Image Processing (ICIP)*, pages 1927–1931. IEEE, 2014. 2

[45] 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 Transactions on Graphics (TOG)*, 26(3):69, 2007. 2

[46] T.-C. Wang, J.-Y. Zhu, N. K. Kalantari, A. A. Efros, and R. Ramamoorthi. Light field video capture using a learning-based hybrid imaging system. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2017)*, 36(4), 2017. 3

[47] S. Wanner and B. Goldluecke. Globally consistent depth labeling of 4D light fields. In *Computer Vision and Pattern Recognition (CVPR)*, pages 41–48. IEEE, 2012. 3

[48] S. Wanner and B. Goldluecke. Reconstructing reflective and transparent surfaces from epipolar plane images. In *German Conference on Pattern Recognition*, pages 1–10. Springer, 2013. 1

[49] S. Wanner, C. Straehle, and B. Goldluecke. Globally consistent multi-label assignment on the ray space of 4D light fields. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2013. 3

[50] G. Wetzstein, I. Ihrke, D. Lanman, and W. Heidrich. Computational plenoptic imaging. In *Computer Graphics Forum*, volume 30, pages 2397–2426. Wiley Online Library, 2011. 2

[51] B. Wilburn, N. Joshi, V. Vaish, E. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, and M. Levoy. High performance imaging using large camera arrays. *ACM Transactions on Graphics (TOG)*, 24(3):765–776, 2005. 2

[52] Y. Xu, K. Maeno, H. Magahara, and R. I. Taniguchi. Camera array calibration for light field acquisition. *Frontiers of Computer Science*, 9(5):691–702, 2015. 3

[53] Y. Xu, H. Nagahara, A. Shimada, and R. I. Taniguchi. Transcut: transparent object segmentation from a light-field image. In *Intl. Conference on Computer Vision (ICCV)*, pages 3442–3450, 2015. 2, 4

[54] Y. Zhang, P. Yu, W. Yang, Y. Ma, and J. Yu. Ray space features for plenoptic structure-from-motion. In *Computer Vision and Pattern Recognition (CVPR)*, pages 4631–4639, 2017. 2

[55] Y. Zhong. Intrinsic shape signatures: A shape descriptor for 3D object recognition. In *Computer Vision (ICCV) Workshops*, pages 689–696. IEEE, 2009. 2