Simple Change Detection from Mobile Light Field Cameras

Donald G. Dansereau[a], Stefan B. Williams[b], Peter I. Corke[a]

[a] *ARC Centre of Excellence for Robotic Vision, Queensland University of Technology Brisbane QLD 4001, Australia* {*donald.dansereau, peter.corke*}*@qut.edu.au*
www.roboticvision.org
[b] *Australian Centre for Field Robotics, School of Aerospace, Mechanical and Mechatronic Engineering, University of Sydney, NSW, Australia s.williams@acfr.usyd.edu.au*

Abstract

Vision tasks are complicated by the nonuniform apparent motion associated with dynamic cameras in complex 3D environments. We present a framework for light field cameras that simplifies dynamic-camera problems, allowing stationary-camera approaches to be applied. No depth estimation or scene modelling is required – apparent motion is disregarded by exploiting the scene geometry implicitly encoded by the light field. We demonstrate the strength of this framework by applying it to change detection from a moving camera, arriving at the surprising and useful result that change detection can be carried out with a closed-form solution. Its constant runtime, low computational requirements, predictable behaviour, and ease of parallel implementation in hardware including FPGA and GPU make this solution desirable in embedded application, e.g. robotics. We show qualitative and quantitative results for imagery captured using two generations of Lytro camera, with the proposed method generally outperforming both naive pixel-based methods and, for a commonly-occurring class of scene, state-of-the-art structure from motion methods. We quantify the tradeoffs between tolerance to camera motion and sensitivity to change, and the impact of coherent, widespread scene changes. Finally, we discuss generalization of the proposed framework beyond change detection, allowing classically still-camera-only methods to be applied in moving-camera scenarios.

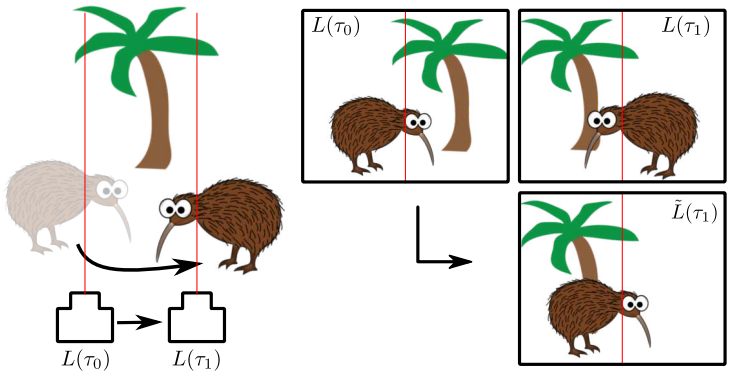


Figure 1: Camera motion between times τ_0 and τ_1 (left) causes apparent motion in static scene elements like the tree (top insets), making them difficult to disambiguate from genuinely dynamic elements, like the Kiwi. We render a novel view $\tilde{L}(\tau_1)$ showing scene content from time τ_0 as seen from the point of view of the camera at τ_1 (bottom right). Static elements now appear static, opening a family of dynamic-camera problems to static-camera solutions. No 3D model of the scene is required, rather the geometry implicitly encoded in the light field is directly exploited. In the case of change detection, this process yields a closed-form solution.

Keywords: Change detection, light field filtering, plenoptic flow, light field rendering

1. Introduction

Having a static camera simplifies a wide range of important computer vision problems: change / motion detection, object tracking, segmentation, isolation and removal, and a range of spatio-temporal filtering techniques including denoising and velocity filtering [1–4]. If the camera is mobile, however, nonuniform apparent motion complicates these techniques, generally requiring structure from motion approaches which generate explicit 3D models of the scene. These methods are conceptually, computationally and behaviourally much more complex than their still-camera counterparts.

We show that light field cameras [5, 6] offer a simplification by allowing a virtual, stationary view to be rendered from a dynamic light field sequence. The process, depicted in Figure 1, does not form an explicit 3D model of the scene. Rather the geometry implicitly encoded in the light field is directly exploited

to produce a virtual, stationary camera, effectively reducing moving-camera problems to stationary-camera problems.

The proposed framework can simplify a wide range of problems. In this work we demonstrate the simplification of change detection, arriving at the surprising, important and novel result that change detection can be carried out with a single closed-form expression. To our knowledge this is the first published closed-form solution to change detection from a moving camera in a 3D environment.

The proposed method outperforms competing single-camera structure from motion approaches for a commonly-occurring class of scene. Because structure from motion jointly estimates camera velocity and scene geometry, changes in the scene can be confused for apparent motion, leading to a significant underestimation of change.

In contrast to competing methods, our solution has constant runtime, low computational requirements, predictable behaviour, and is easily implemented in hardware including FPGA or GPU, making it desirable in a range of challenging application domains including robotics.

The remainder of this paper is organized as follows: We discuss related work in Section 2 and provide background on the closed-form method of camera motion estimation from plenoptic flow in Section 3. We then describe a linear, additive rendering method based on plenoptic flow in Section 4, and combine the methods to effect change detection in Section 5. Section 6 shows results for imagery captured using two generations of Lytro camera, giving quantitative and qualitative analyses of the method's performance and limitations, including explorations of the interplay between sensitivity to change and tolerance to camera motion, and sensitivity to widespread scene changes. The paper concludes with discussion and directions for future work in Section 7, including generalization of the proposed framework over an important class of computer vision problems.

2. Related Work

Change detection from mobile platforms is nontrivial due to the apparent motion of the environment in the captured imagery. This apparent motion is nonuniform in the case of non-planar 3D scene geometry, and so methods based on pixel-level statistics are insufficient for such applications. The key limitation of these techniques is in their direct use of 2D monocular imagery in what is fundamentally a higher-dimensional problem.

Several successful approaches to change detection have been demonstrated under a variety of scene and camera constraints. For sequences with a static camera, the projection of the background onto the image plane is also static, and so it is possible to utilize simple pixel-based statistics to accomplish segmentation [1–3]. This is appealing for several reasons: It is computationally efficient regardless of scene complexity, it is easily parallelized, and it does not rely on identifying and tracking features, which can be problematic in noisy or self-similar environments. Other more sophisticated linear methods are also possible in the case of a stationary camera. For example, the linear velocity filters for object detection proposed in [4]. The work we present is conceptually similar to these filters, but is also applicable when the camera is in motion.

Extension to rotating cameras exploits the lack of parallax in the motion of the background [7–9], and so methods similar to the static-camera case may be employed. Similarly, approximately planar scenes with camera motion parallel to the plane – such as in aerial surveillance – present little or no parallax, and so similar techniques may be employed once the images are registered [10].

In the case of a freely moving camera and nontrivial scene geometry, background elements display different projected velocities. Several approaches have been proposed for addressing this scenario, including the use of occlusion detection, and employing concepts from optical flow to perform iterative camera motion and motion boundary estimation [11, 12].

Other interesting approaches exploit constraints on projected background motion in an orthographic camera, as in [13] which tracks features across the

image sequence, modelling background motion as a sum of basis trajectories. Dense per-pixel labelling is then performed in a final optimization step. In [14] motion between pairs of images is considered, for which background elements are shown to lie on a 1D locus. This constraint is exploited to detect foreground elements, though only low-density results are demonstrated. Dey et al. [15] present a generalization of the epipolar constraint and propose a feature-based approach for exploiting it. Finally, a lightweight algorithm exploiting similar ideas has recently been demonstrated operating in realtime on mobile devices [16].

In a related light field processing paper, Smith et al. [17] render views from a virtual camera with a smoothed trajectory, to effect video stabilization. Our approach differs in rendering views from a stationary virtual camera, allowing change detection to operate simply on a per-pixel basis.

The proposed method requires no feature tracking, no explicit 3D scene model is formed, and no iterative optimization is required. This is behaviourally and computationally simpler than existing methods, and yields results in constant runtime.

This paper builds on the concept of plenoptic flow introduced in [18], introducing a framework for simplifying moving-camera problems, deriving closed-form rendering from plenoptic flow, and providing a simple closed-form expression for change detection. A more detailed treatment can be found in [19].

3. Background: Plenoptic Flow

In this work we employ a relative two-plane parameterization of light rays in which an s, t plane defines ray position, and a u, v plane, closer to the scene at an arbitrary distance D, defines ray direction. In the relative parameterization, u and v are expressed relative to s and t [19]. We employ τ to denote time.

Plenoptic flow and its precursors were first introduced to estimate camera motion [18–20]. This operates much like motion estimation from 2D optical flow [21, 22], but generalizing to six degree-of-freedom (DOF) motion. The equation of plenoptic flow expresses the temporal light field derivative L_τ in

terms of the spatial and angular derivatives L_s, L_t, L_u and L_v, and the camera's translation (q_x, q_y, q_z) and rotation (w_x, w_y, w_z):

$$\begin{bmatrix} L_s \\ L_t \\ \text{-}(uL_s + vL_t)/D \\ \text{-}(tuL_s + tvL_t + uvL_u + v^2L_v)/D - DL_v \\ (suL_s + svL_t + u^2L_u + uvL_v)/D + DL_u \\ sL_t - tL_s + uL_v - vL_u \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} q_x \\ q_y \\ q_z \\ w_x \\ w_y \\ w_z \end{bmatrix} = \text{-}L_\tau, \tag{1}$$

where L_* denotes the partial derivative $\partial L / \partial *$. Partial derivatives are estimated using the first difference.

The equation of plenoptic flow (1) is a linear system, which we can write more compactly as

$$\boldsymbol{A}\boldsymbol{v} = L_\tau. \tag{2}$$

A closed-form least-squares solution to this linear system yields an estimate of the camera's motion $\tilde{\boldsymbol{v}}$ [18, 23] – note that we have absorbed the negation of the temporal derivatives into $\boldsymbol{v}$ to directly yield camera motion. In the following sections we will use this motion estimate to render a novel view which aligns two input light fields.

4. Closed-Form Rendering with Plenoptic Flow

Each of the columns of the matrix $\boldsymbol{A}$ is shown in expanded form in (1). Note that the matrix is shown transposed so that each column is printed as a row, and each of these columns can be interpreted as the change in the light field in response to one of six separate motion components. We will refer to these components as $L_x, L_y, L_z, L_{\omega x}, L_{\omega y}$ and $L_{\omega z}$, respectively. Though they are treated as vectors in solving for camera motion, each of the six components can also be interpreted as a 4D light field, taking on the same dimensions as the input. Taking this approach, we decomposed the light field depicted in Figure 2 into its six motion components, depicted in Figure 3 – negative values are shown as dark, positive as bright, and zero as grey. For these figures, the input was

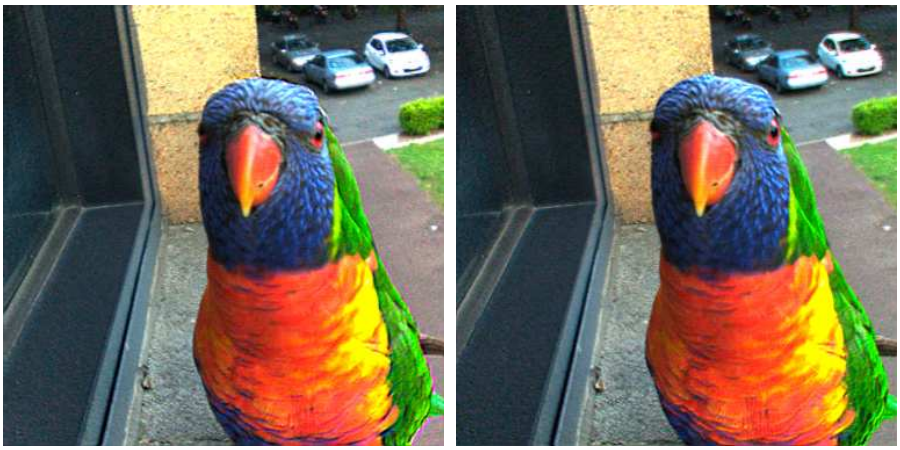


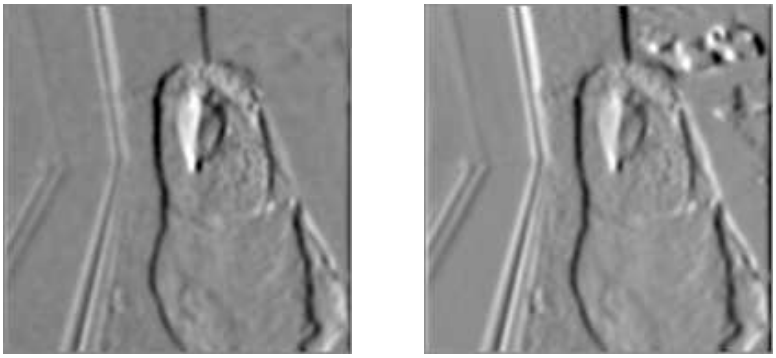
Figure 2: In an effect difficult to capture in print, the rightmost image displays a shifted perspective as accomplished entirely by adding motion components to the input light field – the virtual viewpoint has been translated towards the Lorikeet relative to the measured view, causing the bird to appear larger.

band-limited to a normalized bandwidth of $10^{-0.5}$ to increase the visibility of the derivatives for display.

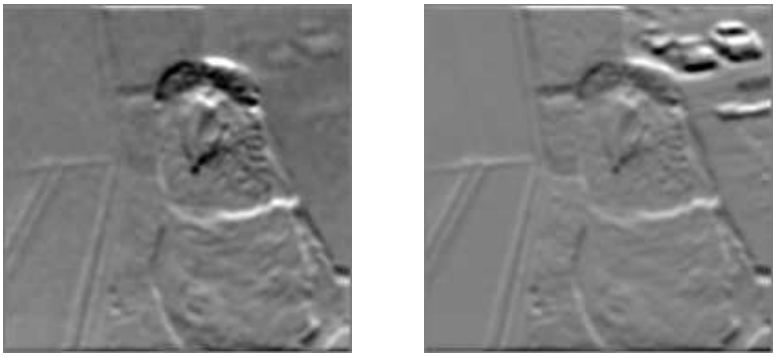
One of the immediate applications of this decomposition is that novel views can now be synthesized via the weighted addition of these six motion components to the original light field, provided the desired camera motion is relatively small. This is difficult to demonstrate in print, given the need for relatively small camera motions, but the two frames in Figure 2 display shifted camera perspectives. The camera has been moved forward in the frame on the right, causing the bird to appear larger, with little change to the more distant background elements. The effect is accomplished entirely through addition of motion components – in this case the displayed light field is the result of adding $8 \times L_z$ to the input light field.

4.1. Motion Ambiguity

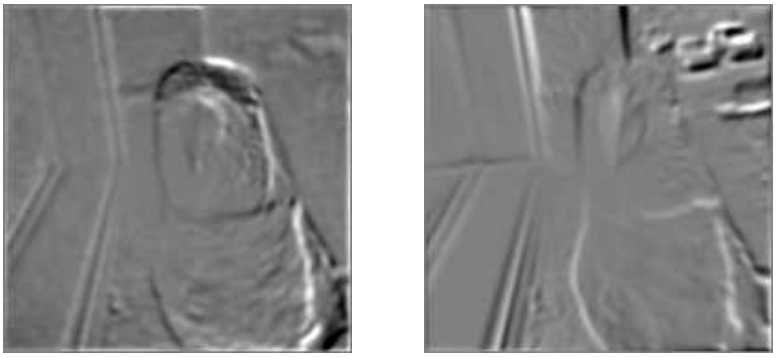
Examining Figure 3, notice that the vertical spatial derivative, L_y and the rotational derivative $L_{\omega x}$ are visually similar, and likewise for L_x and $L_{\omega y}$ – the negation of $L_{\omega x}$ is displayed to emphasize the structural similarity to L_y. This similarity is even more pointed for scenes with less depth variation. In some



(a) L_x (b) $L_{\omega y}$



(c) L_y (d) $-L_{\omega x}$



(e) L_z (f) $L_{\omega z}$

Figure 3: Plenoptic motion components for the scene in Figure 2 – note that angular and spatial derivatives are similar, but not identical.

circumstances, the spatial and rotational derivatives are sufficiently similar that the method of plenoptic flow is unable to distinguish them. This problem has been previously noted [24], and is generally worse in cameras with narrower fields of view, for which the ambiguity is stronger.

Fortunately, though this ambiguity can severely impede motion estimation, it does not significantly impact the rendering of views from a stationary virtual camera. This will be discussed in Section 5.1.

4.2. Rendering Views from a Stationary Virtual Camera

Given two frames, we begin by finding the least squares solution to the equation of plenoptic flow (2) to yield an estimate of the camera's motion $\tilde{\boldsymbol{v}}$. Based on this motion estimate, we wish to render a novel view using the additive method described in Section 4. Again the equation of plenoptic flow gives us the tool to do this, by allowing us to derive the temporal derivative due to the estimated camera motion:

$$\tilde{L}_\tau = \boldsymbol{A}\tilde{\boldsymbol{v}}. \tag{3}$$

Rendering the light field measured at time τ_0 as though viewed from the camera's position at time τ_1 can be accomplished by adding

$$\tilde{L}(\tau_1) = L(\tau_0) + \tilde{L}_\tau. \tag{4}$$

5. Change Detection

Finally, we effect change detection through pixel differencing, by taking the difference between the measured frame $L(\tau_1)$ and the estimated stationary frame $\tilde{L}(\tau_1)$. By substituting (4) and from the definition of the temporal derivative, we find

$$\boldsymbol{R} = L(\tau_1) - \tilde{L}(\tau_1) = L_\tau - \tilde{L}_\tau. \tag{5}$$

In other words, the result of pixel differencing using this method simplifies to the residual error in the equation of plenoptic flow. This is a satisfying result,

as dynamic objects will break the rules underlying plenoptic flow, appearing as areas of high error in the residual. This simple solution is featureless, linear and closed-form.

5.1. Limitations

Rendering views from a stationary virtual camera limits the range of camera motion so that the content of interest remains in-frame. Because we employ plenoptic flow, we introduce the further constraint that camera motion between frames must be small, as in conventional optical flow [21, 22].

It is an elegant result that pixel-wise change detection simplifies to the residual error in plenoptic flow. However, this means that other forms of residual error will also appear as motion. These include occlusions and specular highlights, which break the assumptions underlying the equation of plenoptic flow. Because camera motion between frames is necessarily small, the impact of these effects should be limited. These sources of error should also be easy to detect and ignore – we leave this as future work.

Scenes dominated by dynamic elements can sometimes cause plenoptic flow to describe the dynamic elements' motion rather than the camera's motion, effectively breaking this solution. Changes in illumination will also, as in conventional pixel-wise change detection, cause false positives.

In Section 4.1 we described ambiguities between pairs of rotational and translational motion components within the equation of plenoptic flow. In the present application, we are interested only in identifying elements that break the rules of parallax motion. In this sense, we are not immediately concerned with the velocity estimate $\tilde{\boldsymbol{v}}$, but rather in the reconstructed temporal derivative estimate $\tilde{L}_\tau$ that it yields. As such, ambiguity in the motion components is irrelevant to the task – these components are able to explain the temporal derivative, but not the dynamic scene elements, and so serve our purpose despite the ambiguity in the motion estimate.

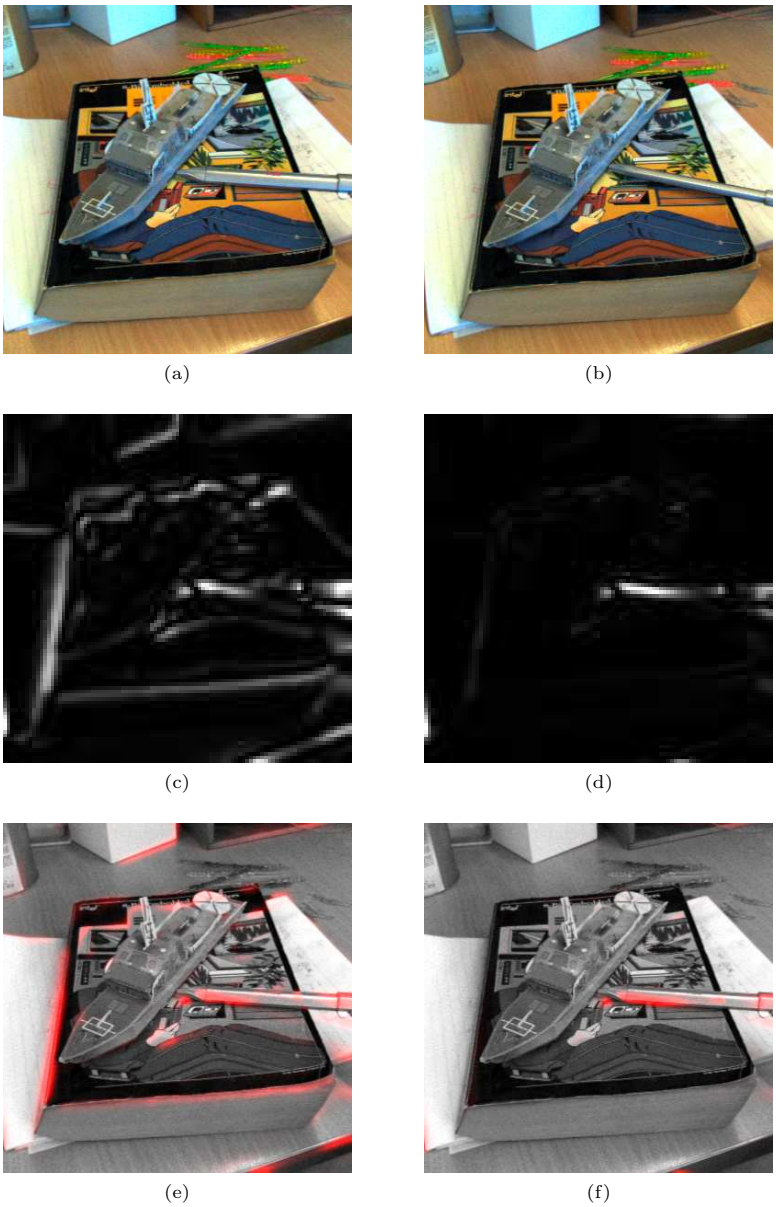
6. Experiments

We applied the method of plenoptic residuals to pairs of images captured using commercially available Lytro consumer-grade plenoptic cameras. The results in the present section were captured with a first-generation Lytro, while those in Sections 6.1 onward were captured using a Lytro Illum second-generation camera. The cameras were calibrated and imagery rectified using the MATLAB Light Field Toolbox [25]. For the Illum, pixels near the edges of lenslets were discarded, as these did not conform well to the simple distortion model employed in [25].

In poorly-lit scenes a hyperfan volumetric focus filter [26] was applied to improve contrast and reject noise, while maintaining depth of field and 3D scene information. We applied a numerically stable form of plenoptic flow, including the method for directly estimating derivatives from rectified light field imagery described in Section 5.3.1 of [19]. Finally, we computed the plenoptic residual (5) to build a map highlighting dynamic scene elements.

The top row of Figure 4 shows two input frames with a small inter-frame camera motion and a single dynamic scene element. The center row shows the magnitude of the difference between frames L_τ, as computed after band-limiting, for plenoptic flow (left), and the plenoptic residual $\boldsymbol{R}$ (right). The bottom row highlights dynamic scene elements in red using L_τ and $\boldsymbol{R}$. The results in Figures 4(c) and (e), representative of naive pixel differencing, show significant sensitivity to apparent motion. Though imperfect, the plenoptic residual results in Figures 4(d) and (f) show significant attenuation of apparent motion, while retaining genuine changes.

Additional results are shown in Figure 5. Each of the three tests captured both dynamic scene elements and nonuniform apparent motion due to a change in camera pose. The left column depicts the result of naive frame differencing, while the right shows the proposed method of plenoptic residuals. Notice the correctly identified shadow change in the first row, and that the two highlights in this row correspond to the original and destination locations of the toothpick in



(a) (b) (c) (d) (e) (f)

Figure 4: Two frames (top) showing both apparent motion and a dynamic scene element. The temporal derivative (c) represents a naive pixel-differencing approach; the plenoptic residual (d) shows significantly less sensitivity to apparent motion while retaining dynamic elements. The first input frame is highlighted using each of these results (bottom). Notice that the pen rotated about its center, thus the pattern of decreasing velocity near its pivot.

Table 1: Energy in the naive pixel difference L_τ and the plenoptic residual R

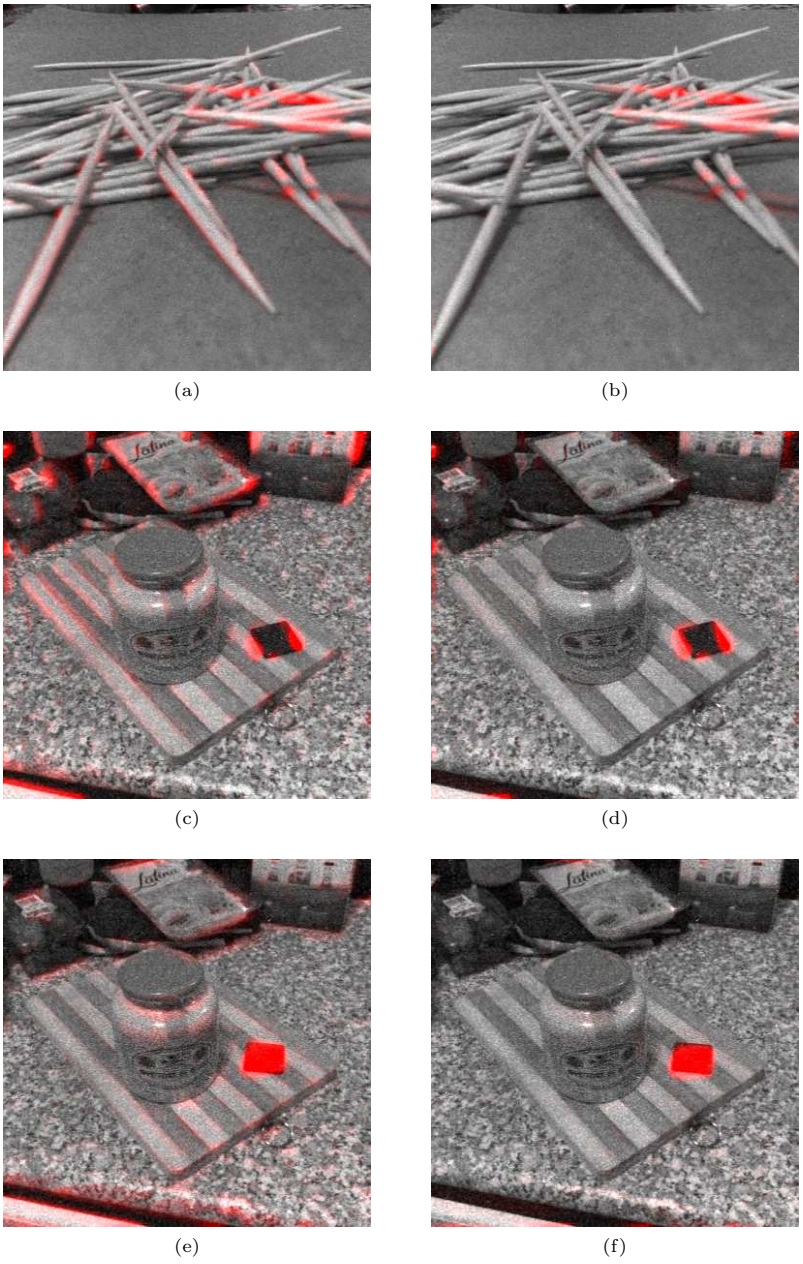
Scene	L_τ (dB)	R (dB)	Ratio (dB)
Jar	-31.81	-35.848	4.0386
Jar	-27.634	-31.029	3.3954
Jar	-36.452	-43.197	6.7448
Pen	-23.805	-28.842	5.037
Pen	-34.679	-39.917	5.2385
Toothpicks	-33.064	-33.55	0.48605
Toothpicks	-30.576	-32.087	1.5104
Toothpicks	-39.247	-42.276	3.0284
Mean	-29.684	-33.439	4.0905

a relatively large translation. In the bottom row, the square object was removed between frames, while in the center row it was rotated.

Table 1 summarizes the signal energy resulting from naive pixel differencing and the method of plenoptic residuals, and their ratio. Values are shown for eight pairs of images from the three test scenes depicted in Figures 4 and 5. The tabulated values represent signal energy expressed in dB, for input light fields normalized to a peak value of one. The mean ratio of 4 dB establishes that the plenoptic residuals method is more than twice as selective as naive pixel differencing. Referring to Figures 4 and 5, we confirm the method has selectively attenuated static scene elements while passing dynamic objects.

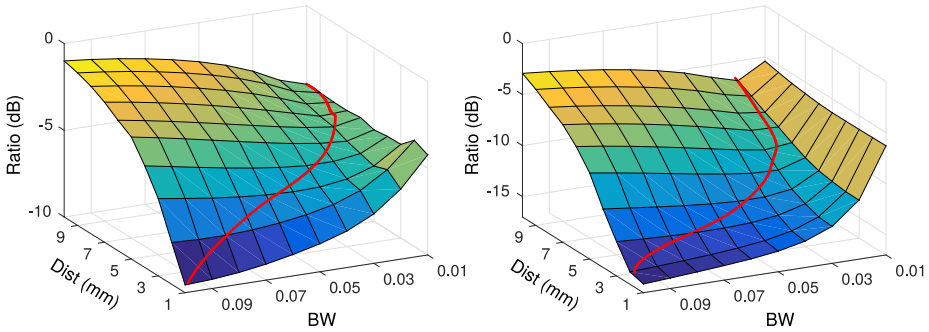
6.1. Tolerance to Camera Motion

One of the limitations of the proposed method is that camera motion between frames must be small. There is an interplay between input bandwidth, sensitivity to change, and tolerance to camera motion. To demonstrate this we measured the performance of plenoptic residuals over a range of camera motion magnitudes for a range of input bandwidths. A first, fixed frame was compared with a series of frames showing increasingly more camera motion. The camera was translated along x from 1 to 10 mm in increments of 1 mm. The test was run on a static scene, shown in Figure 6(c), and on a dynamic scene, for which the second and subsequent frames had a change as seen in Figure 6(d).



(a) (b) (c) (d) (e) (f)

Figure 5: Additional results demonstrating the method of plenoptic residuals – the left column demonstrates naive temporal differencing, while the right demonstrates the proposed method.



(a) Static scene performance

(b) Dynamic scene performance

(c) Static scene

(d) Scene after change

Figure 6: Performance varies with camera motion and input bandwidth (BW) for (a) the static scene shown in (c); and (b) the same scene with a change introduced, as shown in (d). In both plots lower values represent better performance, and the red path highlights optimal bandwidth as a function of camera motion. For (a) performance is shown as the ratio of false positive change detection to naive pixel differencing, and for (b) as the ratio of false positive to true positive change detection. Note that to tolerate larger camera motions the input bandwidth must be decreased, but this limits sensitivity to change as manifested to the right in (b). Each plot represents the mean over two experiments.

Experiments for both static and dynamic-scenes were repeated twice, with strong agreement between experiments. The average results are shown in Figure 6. In the case of the static scene, performance was evaluated as the ratio of false positive change detected to the estimate yielded by naive pixel differencing. Because the scene was static, the ideal result is that no change be detected, and so lower values are better. Notice that for higher camera displacements the optimal input bandwidth, highlighted in red, is lower. This is because the coherence of the input must be increased to tolerate larger shifts.

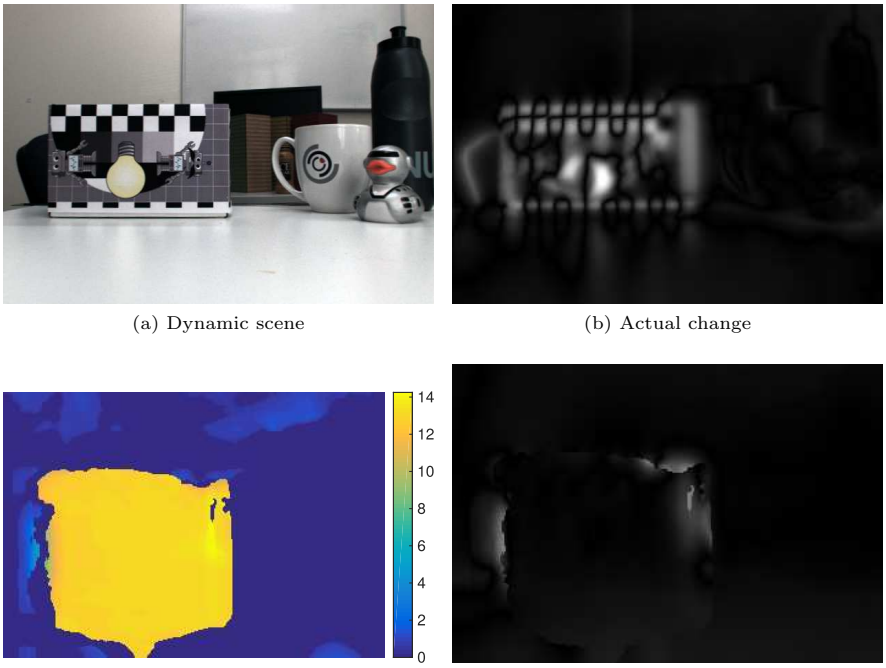
For the dynamic scene, hand-labelled ground truth was used to find the ratio between false positive change detection and true positive change detection – again, lower values are better. The shape of the result, shown in Figure 6(b), is similar to the static case, except for a more prominent decrease in performance for very small bandwidths. This is due to a decrease in sensitivity to change, which does not appear in the static experiment.

6.2. Comparison to Structure from Motion

As discussed in Section 2, most competing change detection methods are either sparse or much more complex than the proposed method. Indeed, in single-camera scenarios change detection generally requires joint estimation of the scene geometry and the camera's motion, which can only be accomplished using sophisticated, iterative optimization methods [27, 28]. By estimating scene geometry and camera motion, two views of the scene can be aligned, and a difference computed to identify dynamic elements.

The depth information implicitly captured by the light field confers two advantages: 1) it allows simplification of change detection to a single-step, closed-form solution, with no explicit geometry estimation required, and 2) the resulting method is robust to an important failure mode common to most if not all competing single-camera techniques.

When elements of the scene move with a projected velocity consistent with apparent motion due to the camera's velocity, they can appear as stationary



(a) Dynamic scene

(b) Actual change

(c) Disparity estimate

(d) Estimated change

Figure 7: State-of-the-art single-camera methods fail for projected motion parallel to apparent motion. Here camera motion is constrained to translation along x, reducing depth estimation to stereo matching. The scene (a) has a dynamic element translating to the right, as seen from the temporal derivative (b), but this motion is misinterpreted as disparity due to depth (c), resulting in an incorrect change estimate (d).

objects at shifted depths. We demonstrate this effect in Figure 7, and quantify it in Figure 8.

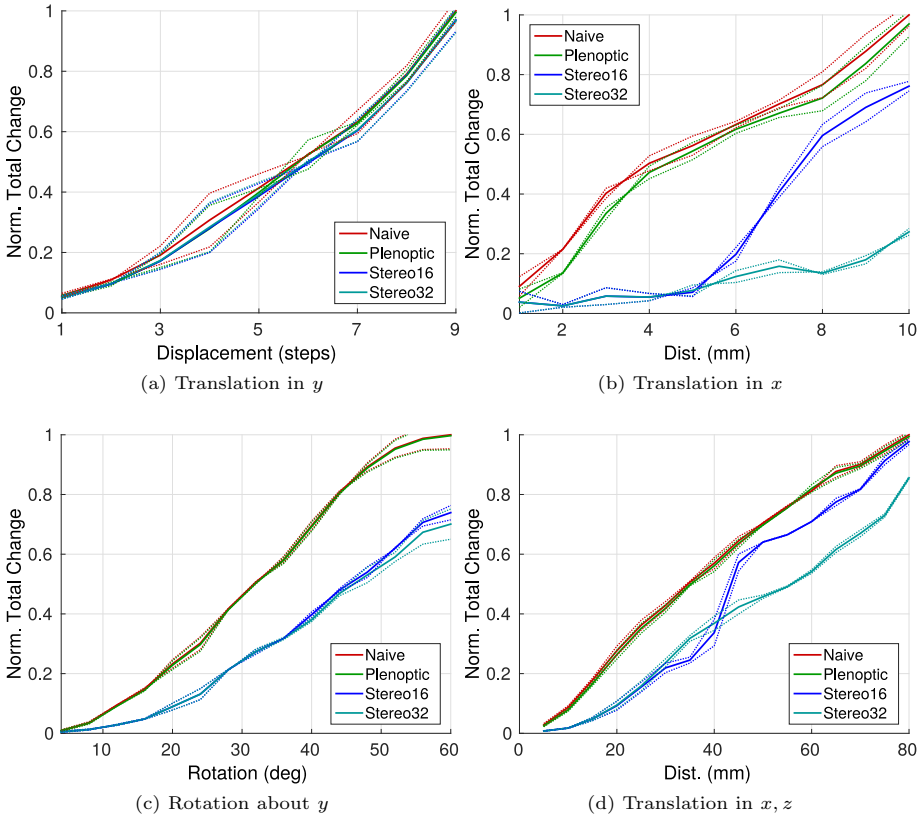
Although the effect applies to dense structure from motion in general, we restrict our attention to the case of a camera moving with known velocity along x. Under these circumstances, dense structure from motion simplifies to dense stereo matching. We estimate disparity using a semi-global block matching approach [29], and use this disparity to reproject the first frame as though seen from the point of view of the second frame. Because this simplified approach is a subset of structure from motion with fewer sources of error, we employ it as an upper bound of performance over more general approaches.

The scene shown Figure 7(a) includes an object moving horizontally along x. The temporal derivative for a horizontal shift of 4 mm, as seen from a stationary camera, is shown in (b). Figure 7(c) shows the disparity estimate, in which the motion of the dynamic object has yielded an overestimation of disparity. The resulting change estimate, shown in (d), shows how the motion of the test pattern has been underestimated, as it has been misinterpreted as depth.

To better understand this failure mode, we tested a variety of motion types and compared the change estimates from naive pixel differencing, plenoptic residuals, and the stereo-based approach. An important parameter of the stereo approach is the maximum disparity, which we tested at 16 and 32 pixels, with the minimum fixed at 0.

Four experiments were run over two repetitions each, with results shown in Figure 8. In each experiment a fixed frame was compared to a series of frames showing increasing motion. Because the camera was fixed, the naive temporal derivative acts as ground truth. All traces are normalized to the maximum temporal derivative, and the heavy lines indicate the mean over the two repetitions shown as dashed lines.

The scene for Figure 8(a) showed only vertical motion and acts as a control, verifying that the stereo-based change detection method operates well for projected changes orthogonal to apparent motion. The scene for (b) showed



(a) Translation in y

(b) Translation in x

(c) Rotation about y

(d) Translation in x, z

Figure 8: Evaluating state-of-the-art single-camera methods. The camera is fixed so that naive pixel differencing represents the ground truth, and stereo matching represents an upper bound on the performance of structure from motion approaches. (a) For vertical scene motion, all methods perform well, while stereo methods suffer for scenes showing (b) translation along x, (c) rotation about y, or (d) diagonal translation in x, z. The larger motions in (b) and (d) exceed the maximum 16-pixel disparity of *Stereo16*, causing an increase in performance compared with *Stereo32*. Heavy lines indicate the mean over two repetitions, shown as dashed lines.

translation of the test pattern to the right along x, while (c) showed rotation about y, i.e. with projected motion to the right, and (d) showed diagonal motion to the right and away from the camera at about a 45 degree angle in x, z.

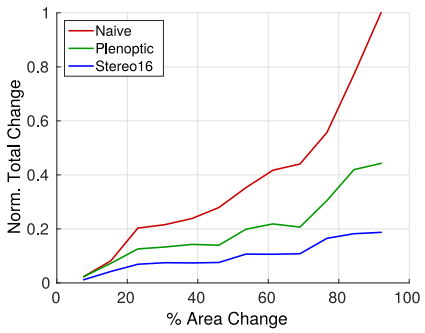
Plenoptic residuals performed well across all four motion types, while the stereo method showed poor performance for examples with horizontal projected motion. Note that fixing a maximum stereo disparity of 16 yielded improved results where the projected motion exceeded 16 pixels, as can be seen in Figures 8(b) and (d), but not in (c), for which projected motion did not exceed 16 pixels.

6.3. Maximum Scene Motion

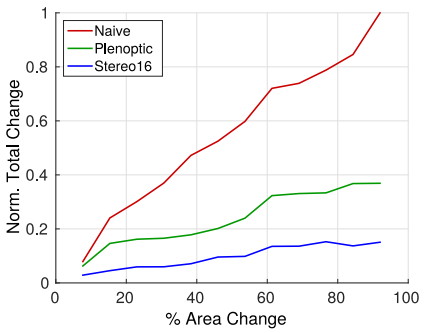
When motion dominates a scene it can be confused for apparent motion, causing dynamic elements to be misinterpreted as being static. To demonstrate this we constructed a scene out of 12 movable tiles, and compared the change estimates for naive pixel differencing, plenoptic residuals, and the stereo-based method described above. A first, fixed frame was compared to a series of frames in which tiles were displaced one at a time. Again the camera was fixed, so that naive pixel differencing represents the ground truth.

For the first two experiments, shown in Figures 9(a) and (b), tiles were translated one at a time, 1 mm to the right, whereas in 9(c) and (d) they were rotated randomly by a few degrees. For the random rotations plenoptic residuals performed well, while for the translations it did not, showing decreasing estimates as more of the scene moved. The coherent translation of the tiles was consistent with apparent motion in that it could have been caused by camera displacement, while the random rotations were not, and so only the former caused a loss in performance.

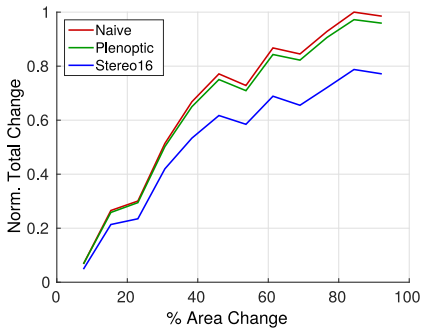
When scene dynamics are consistent with apparent motion it should be possible to fool plenoptic flow into believing there is no scene change whatsoever. To prove this we performed a simulation, depicted in Figure 9(e), in which an increasing percentage of the scene was artificially shifted. The experiment was repeated over a range of shift magnitudes between 1 and 10 pixels. For small



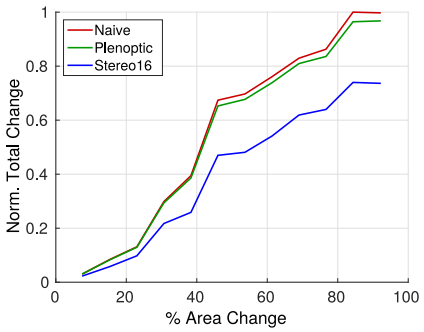
(a) Coherent scene motion



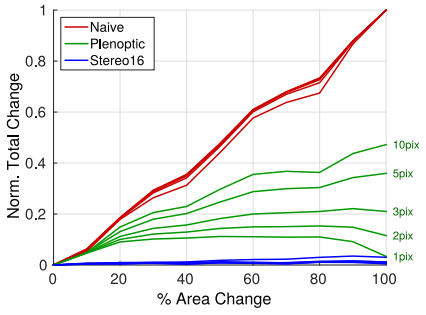
(b) Coherent scene motion



(c) Random scene motion



(d) Random scene motion



(e) Simulated coherent motion

Figure 9: (a),(b) Coherent, small horizontal motion across the scene yields poor results as plenoptic flow is fooled into thinking the camera has moved rather than the scene; (c),(d) random scene motion does not cause the same problem, as it cannot be explained by camera motion; (e) Simulated coherent motion, with motion ranging from 1 to 10 pixels in magnitude: Small motion across the entire frame yields the worst results. Stereo performs poorly throughout these experiments, as explained in the previous section.

pixel shifts of the whole image, all motion was interpreted as camera motion, yielding a plenoptic residual near zero. Larger shifts showed less dramatic results, however. This is consistent with the results shown in Figure 6: Large motions beyond the coherence of the scene break plenoptic flow, and so rather than being misinterpreted as camera motion, these were are at least partially interpreted as scene motion.

Note that throughout these experiments, the stereo-based method significantly underestimated change due to the presence of horizontal motion, even in the case of random tile rotations as depicted in Figures 9(c) and (d).

7. Discussion and Future Directions

We presented a general approach for converting moving-camera problems into stationary-camera problems. No depth estimation or other complex scene modelling is required – apparent motion is disregarded by directly exploiting the geometric information implicitly encoded by the light field.

Using this approach, we derived a method for closed-form change detection from moving platforms. By effecting both camera motion estimation and rendering using closed-form plenoptic flow, we showed that pixel-wise change detection from a virtual still camera can be found from the residual error in plenoptic flow. This is an important, surprising and useful result: Its constant runtime, low computational requirements, predictable behaviour, and ease of parallel implementation in hardware including FPGA and GPU make it desirable for deployment in demanding embedded applications including robotics.

We evaluated the method of plenoptic residuals using first- and second-generation Lytro cameras. We showed the method to outperform naive 2D per-pixel methods, which are sensitive to nonuniform apparent motion of the scene, and sophisticated structure from motion approaches, for the important case of projected motion parallel with apparent motion due to the camera's velocity. We quantified the tradeoff between tolerance to camera motion and

sensitivity to change, and the susceptibility of the proposed method to coherent, widespread scene movement.

As future work it should be possible to derive a more conventional approach that nevertheless exploits the depth information captured by light field cameras. For example, pairs of virtual views could be rendered for each camera pose and employed in stereo structure from motion. We expect such approaches to show less sensitivity to apparent motion than their monocular counterpart, but that plenoptic residuals will remain behaviourally and computationally simpler and therefore attractive for hardware implementation and embedded deployment.

The method of plenoptic residuals is susceptible to false positives where the assumptions underlying plenoptic flow are broken. These include occlusions, specular reflections, and changes in illumination. A method of detecting and explicitly ignoring these phenomena would be desirable, both in change detection and in improving the performance of plenoptic flow-based visual odometry.

Finally, this work started with a framework to efficiently and linearly co-register light field images to simplify a class of computer vision problems. We leave as future work demonstration on other problems in this class, including object tracking, segmentation, isolation and removal, and a range of spatio-temporal filtering techniques including denoising and velocity filtering [1–4].

8. Acknowledgments

This work was supported by the Australian Centre for Field Robotics (Project DP150104440) and the Australian Research Council Centre of Excellence for Robotic Vision (Project CE140100016). Thanks to the reviewers, Dr. Linda Miller and Dr. Jürgen Leitner for their helpful suggestions.

References

[1] M. Piccardi, Background subtraction techniques: a review, in: IEEE Intl. Conference on Systems, Man and Cybernetics, Vol. 4, 2004, pp. 3099–3104.

[2] S. Chien, S. Ma, L. Chen, Efficient moving object segmentation algorithm using background registration technique, IEEE Transactions on Circuits and Systems for Video Technology 12 (7) (2002) 577–586.

[3] C. Stauffer, W. Grimson, Adaptive background mixture models for real-time tracking, in: Computer Vision and Pattern Recognition (CVPR), Vol. 2, IEEE, 1999, pp. 246–252.

[4] S. Schauland, J. Velten, A. Kummert, Detection of moving objects in image sequences using 3D velocity filters, Intl. Journal of Applied Mathematics and Computer Science 18 (1) (2008) 21–31.

[5] M. Levoy, P. Hanrahan, Light field rendering, in: SIGGRAPH, ACM, 1996, pp. 31–42.

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

[7] E. Hayman, J. Eklundh, Statistical background subtraction for a mobile observer, in: Intl. Conference on Computer Vision (ICCV), 2003, pp. 67–74.

[8] A. Mittal, D. Huttenlocher, Scene modeling for wide area surveillance and image synthesis, in: Computer Vision and Pattern Recognition (CVPR), Vol. 2, IEEE, 2000, pp. 160–167.

[9] Y. Ren, C. Chua, Y. Ho, Statistical background modeling for non-stationary camera, Pattern Recognition Letters 24 (1-3) (2003) 183–196.

[10] R. Pless, T. Brodsky, Y. Aloimonos, Detecting independent motion: the statistics of temporal continuity, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) 22 (8) (2000) 768–773.

[11] A. Ogale, C. Fermuller, Y. Aloimonos, Motion segmentation using occlusions, IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) 27 (6) (2005) 988–992.

[12] R. Feghali, A. Mitiche, Spatiotemporal motion boundary detection and motion boundary velocity estimation for tracking moving objects with a moving camera: a level sets PDEs approach with concurrent camera motion compensation, IEEE Transactions on Image Processing (TIP) 13 (11) (2004) 1473–1490.

[13] Y. Sheikh, O. Javed, T. Kanade, Background subtraction for freely moving cameras, in: Intl. Conference on Computer Vision (ICCV), 2009, pp. 1219–1225.

[14] R. C. Nelson, Qualitative detection of motion by a moving observer, Intl. Journal of Computer Vision (IJCV) 7 (1991) 33–46.

[15] S. Dey, V. Reilly, I. Saleemi, M. Shah, Detection of independently moving objects in non-planar scenes via multi-frame monocular epipolar constraint, in: European Conference on Computer Vision (ECCV), 2012, pp. 860–873.

[16] K. Yi, K. Yun, S. W. Kim, H. J. Chang, H. Jeong, J. Y. Choi, Detection of moving objects with non-stationary cameras in 5.8ms: Bringing motion detection to your mobile device, in: Mobile Vision, 3rd IEEE Intl. Workshop on, IEEE, 2013, pp. 27–34.

[17] B. Smith, L. Zhang, H. Jin, A. Agarwala, Light field video stabilization, in: Intl. Conference on Computer Vision (ICCV), IEEE, 2010, pp. 341–348.

[18] D. G. Dansereau, I. Mahon, O. Pizarro, S. B. Williams, Plenoptic flow: Closed-form visual odometry for light field cameras, in: Intelligent Robots and Systems (IROS), IEEE, 2011, pp. 4455–4462.

[19] D. G. Dansereau, Plenoptic signal processing for robust vision in field robotics, Ph.D. thesis, Australian Centre for Field Robotics, School of Aerospace, Mechanical and Mechatronic Engineering, The University of Sydney (Jan. 2014).

[20] J. Neumann, C. Fermuller, Y. Aloimonos, V. Brajovic, Compound eye sensor for 3D ego motion estimation, in: Intelligent Robots and Systems (IROS), Vol. 4, IEEE, 2005, pp. 3712–3717.

[21] B. Lucas, T. Kanade, An iterative image registration technique with an application to stereo vision, in: Intl. Joint Conference On Artificial Intelligence, Vol. 3, 1981, pp. 674–679.

[22] D. Fleet, Y. Weiss, Optical flow estimation, in: Handbook of Mathematical Models in Computer Vision, Springer, 2006, pp. 237–257.

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

[24] J. Neumann, Computer vision in the space of light rays: plenoptic video geometry and polydioptric camera design, Ph.D. thesis, University of Maryland (2004).

[25] D. G. Dansereau, O. Pizarro, S. B. Williams, Decoding, calibration and rectification for lenselet-based plenoptic cameras, in: Computer Vision and Pattern Recognition (CVPR), IEEE, 2013, pp. 1027–1034.

[26] D. G. Dansereau, O. Pizarro, S. B. Williams, Linear volumetric focus for light field cameras, ACM Transactions on Graphics (TOG), Presented at SIGGRAPH 2015 34 (2) (2015) 15.

[27] R. A. Newcombe, S. Lovegrove, A. J. Davison, DTAM: dense tracking and mapping in real-time, in: Intl. Conference on Computer Vision (ICCV), 2011, pp. 2320–2327.

[28] R. Newcombe, A. Davison, Live dense reconstruction with a single moving camera, in: Computer Vision and Pattern Recognition (CVPR), IEEE, 2010, pp. 1498–1505.

[29] H. Hirschmüller, Accurate and efficient stereo processing by semi-global matching and mutual information, in: Computer Vision and Pattern Recognition (CVPR), Vol. 2, IEEE, 2005, pp. 807–814.