Depth from Focus with Your Mobile Phone

Supasorn Suwajanakorn[1,3], Carlos Hernandez[2] and Steven M. Seitz[1,2]

[1]University of Washington [2]Google Inc.

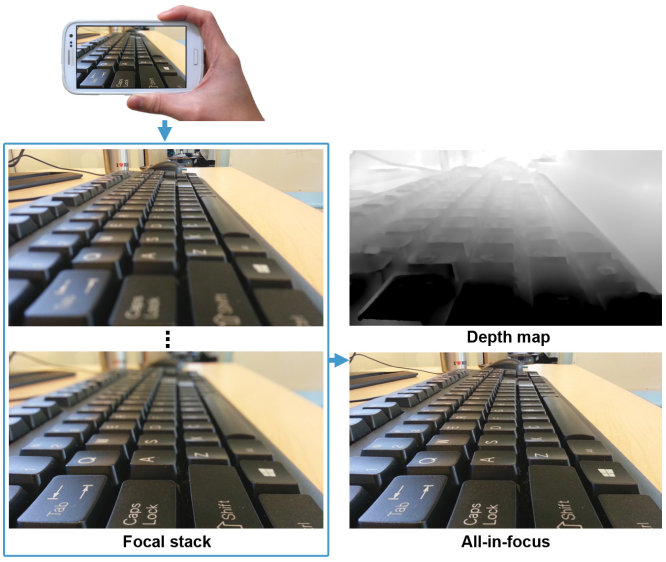

Figure 1. We compute depth and all-in-focus images from the focal stack that mobile phones capture each time you take a photo.

Abstract

While prior depth from focus and defocus techniques operated on laboratory scenes, we introduce the first depth from focus (DfF) method capable of handling images from mobile phones and other hand-held cameras. Achieving this goal requires solving a novel uncalibrated DfF problem and aligning the frames to account for scene parallax. Our approach is demonstrated on a range of challenging cases and produces high quality results.

1. Introduction

Every time you take a photo with your mobile phone, your camera rapidly sweeps the focal plane through the scene to find the best auto-focus setting. The resulting set of images, called a *focal stack*, could in principle be used to compute scene depth, yielding **a depth map for every photo you take**. While depth-from-focus (DfF) techniques have been studied for a couple decades, they have been relegated to laboratory scenes; no one has ever demonstrated an approach that works on standard mobile phones, or other hand-held consumer cameras. This paper presents the first successful demonstration of this capability, which we call *hand-held DfF*.

Hand-held DfF is challenging for two reasons. First, while almost all DfF methods require calibrated capture, supporting commodity mobile phones requires working in an uncalibrated setting. We must solve for the focal settings as part of the estimation process. Second, capturing a focal sweep with a hand-held camera inevitably produces motion parallax. The parallax is significant, as the camera motion is typically on the order of the aperture size (which is very small on a cell phone). I.e., the parallax is often larger than the defocus effect (bokeh radius). Almost all previous DfF methods used special optics (e.g., [28]) to avoid motion, or employed simple global transformations to align images. Only [24] attempts to handle dynamic scenes, which can exhibit parallax, but requires a calibrated camera in the lab. By exploiting a densely-sampled focal stack, we propose a simpler alignment technique based on flow concatenation, and are the first to demonstrate uncalibrated DfF on hand-held devices.

We address the hand-held DfF problem in two steps: 1) focal stack alignment, and 2) auto calibration and depth recovery. The first step takes as input a focal sweep from a moving camera, and produces as output a stabilized sequence resembling a constant-magnification, parallax-free sequence taken from a telecentric camera [28]. Because defocus effects violate the brightness constancy assumption, the alignment problem is especially challenging, and we introduce an optical flow concatenation approach with special handling of highlight (bokeh) regions to achieve high quality results.

In the second step, given an aligned focal stack, we aim to recover the scene depth as well as aperture size, focal length of the camera, and focal distances of each frame up to

[3]This research was done while the first author was an intern at Google.

an affine ambiguity in the inverse depth [17]. To solve this problem, we first compute an all-in-focus photo using an MRF-based approach. Then we formulate an optimization problem that jointly solves for camera settings and scene depth that best explains the focal stack. Finally, we refine the depthmap by fixing the global estimates and solve for a new depthmap with a robust anisotropic regularizer which optimizes surface smoothness and depth discontinuity on occlusion boundaries.

2. Related Work

Almost all prior work for estimating depth using focus or defocus assumes known calibration parameters and parallax-free, constant-magnification input sequences such as ones taken by a telecentric camera [29].

The only previous work relating to uncalibrated DfF is Lou et al. [17], who proved that in the absence of calibration parameters, the reconstruction as well as the estimation of the focal depths will be up to an affine transformation of the inverse depth. Zhang et al. [33] considered a related uncalibrated defocus problem for the special case where only the aperture changes between two images. However, no algorithm has been demonstrated for the case of unknown aperture and focal length or for jointly calibrating focal depths in a focal stack of more than two frames (the case for hand-held DfF).

Prior work exists for geometrically aligning frames in a calibrated image sequence: [11, 34] used an image warping approach to correct for magnification change. [3] propose a unified approach for registration and depth recovery that accounts for misalignment between two input frames under a global geometric transformation. However, none of these techniques address parallax, and therefore fail for hand-held image sequences. Recent work [24] attempts to handle parallax and dynamic scenes by alternating between DfD and flow estimation on reblurred frames, but requires a calibrated camera in the lab.

Instead of sweeping the focal plane, other authors have proposed varying aperture [4, 12, 21, 26, 31]. While this approach avoids magnification effects, it does not account for parallax induced by hand-held capture. Furthermore, since defocus effects are less pronounced when varying aperture size compared to varying focal depths [28], the aperture technique is less applicable to small aperture devices such as mobile phones. A third approach is to fix focus and translate the object [23, 19]. The image sequence produced from this scheme has a constant magnification but exhibits motion parallax along the optical axis. [23] proposes an MRF-based technique to address this kind of parallax, but is not applicable to the more general parallax caused by hand-held camera shake.

3. Overview

One way to solve the problem of estimating the 3D surface from an uncalibrated focal stack (DfF) is to jointly solve for all unknowns, i.e., all camera intrinsics, scene depth and radiance, and the camera motion. The resulting minimization turns out to be intractable and one would need a good initialization near the convex basin of the global minimum for such non-linear optimization. In our case, the availability of the entire focal stack, as opposed to two frames usually assumed in depth-from-defocus problem, enables a relatively simple estimation scheme for the scene radiance. Thus, we propose a technique that first aligns every frame to a single reference (Section 4) and produces an all-in-focus photo as an approximation to the scene radiance (Section 5). With the scene radiance fixed and represented in a single view, the remaining camera parameters and scene depth can then be solved in a joint optimization that best reproduces the focal stack (Section 6).

In addition, we propose a refinement scheme to improve depth map accuracy by incorporating spatial smoothness (Section 6.2) and an approach to correct the bleeding problem for saturated, highlight pixels, known as *bokeh*, in the estimation of an all-in-focus image (Section 7.1). The following sections describe each component in detail.

4. Focal Stack Alignment

The goal of the alignment step is to compensate for parallax and viewpoint changes produced by a moving, hand-held capture. That is, the aligned focal stack should be equivalent to a focal stack captured with a static, telecentric camera.

Previous work corrected for magnification changes through scaling and translating [11] or a similarity transform [3]. However these global transformations are inadequate for correcting local parallax. Instead, we propose a solution based on optical flow which solves for a dense correction field. One challenge is that defocus alters the appearance of each frame differently depending on the focus settings. Running an optical flow algorithm between each frame and the reference may fail as frames that are far from the reference in the focal stack appear vastly different. We overcome this problem by concatenating flows between consecutive frames in the focal stack, which ensures that defocus between two input frames to the optical flow appear similar.

Given a set of frames in a focal stack $I_1, I_2, \ldots, I_n$, we assume without loss of generality that I_1 is the reference frame, which has the largest magnification. Our task is to align $I_2, \ldots, I_n$ to I_1.

Let the 2D flow field that warps I_i to I_j be denoted by $\mathcal{F}_i^j : \mathbb{R}^2 \rightarrow \mathbb{R}^2$. Let $\mathcal{W}_{\mathcal{F}}(I)$ denote the warp of image I

according to the flow $\mathcal{F}$

$$\mathcal{W}_{\mathcal{F}}(I(u,v)) = I(u + \mathcal{F}(u,v)_x, v + \mathcal{F}(u,v)_y), \quad (1)$$

where $\mathcal{F}(u,v)_x, \mathcal{F}(u,v)_y$ are the x- and y-components of the flow at position (u,v) in image I. We then compute the flow between consecutive frames $\mathcal{F}_2^1, \mathcal{F}_3^2, \ldots, \mathcal{F}_n^{n-1}$, and recursively define the flow that warps each frame to the reference as $\mathcal{F}_i^1 = \mathcal{F}_i^{i-1} \circ \mathcal{F}_{i-1}^1$, where $\circ$ is a concatenation operator given by $\mathcal{F} \circ \mathcal{F}' = \mathcal{S}$, $\mathcal{S}_x = \mathcal{F}'_x + \mathcal{W}_{\mathcal{F}'}(\mathcal{F}_x)$ and similarly $\mathcal{S}_y = \mathcal{F}'_y + \mathcal{W}_{\mathcal{F}'}(\mathcal{F}_y)$. Here, we treat $\mathcal{F}_x, \mathcal{F}_y$ as images and warp them according to flow $\mathcal{F}'$. After this step, we can produce an aligned frame $\hat{I}_i = \mathcal{W}_{\mathcal{F}_i^1}(I_i)$.

In the ideal case, the magnification difference will be corrected by the flow. However, we found that computing a global affine transform between I_i and I_{i+1} to compensate for magnification changes or rolling-shutter effects before computing the flow $\mathcal{F}_{i+1}^i$ helps improve the alignment in ambiguous, less-textured regions. Specifically, we compute the affine warp using the Inverse Compositional Image Alignment algorithm [2], and warp I_{i+1} to I_i before computing optical flow.

5. All-in-Focus Image Stitching

Given an aligned focal stack $\hat{I}_1, \hat{I}_2, \ldots, \hat{I}_n$, an all-in-focus image can be produced by stitching together the sharpest in-focus pixels across the focal stack. Several measures of pixel sharpness have been proposed in the shape-from-focus literature [15, 18, 27, 15, 13]. Given a sharpness measure, we formulate the stitching problem as a multi-label MRF optimization problem on a regular 4-connected grid where the labels are indices to each frame in the focal stack. Given $\mathcal{V}$ as the set of pixels and $\mathcal{E}$ as the set of edges connecting adjacent pixels, we seek to minimize the energy:

$$E(x) = \sum_{i \in \mathcal{V}} E_i(x_i) + \lambda \sum_{(i,j) \in \mathcal{E}} E_{ij}(x_i, x_j) \quad (2)$$

where λ is a weighting constant balancing the contribution of the two terms. The unary term $E_i(x_i)$ measures the amount of defocus and is defined as the sum of $\exp|\nabla I(u,v)|$ over a Gaussian patch with variance (μ^2, μ^2) around the pixel (u,v).

The pairwise term, $E_{ij}(x_i, x_j)$ is defined as the total variation in the frame indices $|x_i - x_j|$, which is submodular and can be minimized using the α-expansion algorithm [6, 14, 5].

6. Focal Stack Calibration

Given an aligned focal stack $\hat{I}_1, \hat{I}_2, \ldots, \hat{I}_n$, we seek to estimate the focal length of the camera F, the aperture of the lens A, the focal depth of each frame in the stack $f_1, \ldots, f_n$ and a depthmap representing the scene $s : \mathbb{R}^2 \to [0, \infty)$. We assume that the scene is Lambertian and is captured by a camera following a thin-lens model. While our implementation uses a uniform disc-shaped point spread function (PSF), our approach supports any type of PSF.

Let the radiance of the scene, projected onto the reference frame, be $r : \mathbb{R}^2 \to [0, \infty)$. Each image frame in the radiance space can be approximated by:

$$\hat{I}_i(x,y) = \iint r_{uv}\, D(x-u, y-v, b_i(s_{uv}))\, \mathrm{d}u\, \mathrm{d}v, \quad (3)$$

where $D : \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}$ is a disc-shaped PSF centered at the origin with radius $b_i(s)$ given by:

$$b_i(s) = A \cdot \frac{|f_i - s|}{s} \cdot \frac{F}{f_i - F}, \quad (4)$$

where A is the aperture size and F is the focal length.

One way to calibrate the focal stack is to first approximate scene radiance r_{uv} by the recovered all-in-focus image, $\hat{I}_0$, then re-render each frame using the given focal depths and depthmap:

$$\tilde{I}_i(x,y) = \iint \hat{I}_0(u,v)\, D(x-u, y-v, b_i(x,y))\, \mathrm{d}u\, \mathrm{d}v \quad (5)$$

The total intensity differences between the re-rendered frames and real images can then be minimized across the focal stack with a non-linear least-squares formulation:

$$\min_{A,s,F,f_1,\ldots,f_n} \sum_{i=1}^{n} \left\| \hat{I}_i - \tilde{I}_i(A, s, f_i) \right\|_F^2 \quad (6)$$

This optimization, however, is highly non-convex and involves re-rendering the focal stack thousands of times, which makes convergence slow and susceptible to local minima. Therefore, for calibration purposes, we assume that depth values in the window around each pixel are locally constant, i.e., blur is locally shift-invariant. This assumption allows us to evaluate the rendered result of each pixel by a simple convolution. Moreover, we can now pre-render a blur stack where each frame corresponds to a blurred version of the all-in-focus image for a given PSF radius. We can now formulate a novel objective that is tractable and jointly optimizes all remaining parameters.

6.1. Joint Optimization

Given the assumption that blur is locally shift-invariant, we can generate a blur stack $\hat{I}_0^r$, where each frame in the stack corresponds to the all-in-focus image $\hat{I}_0$ blurred by a constant disc PSF with a fixed radius r. In practice, we generate a stack with blur radius increasing by 0.25 pixels between consecutive frames. The optimization problem can now be formulated as, for each pixel in each frame of the aligned stack, select a blur radius (i.e., a frame in the

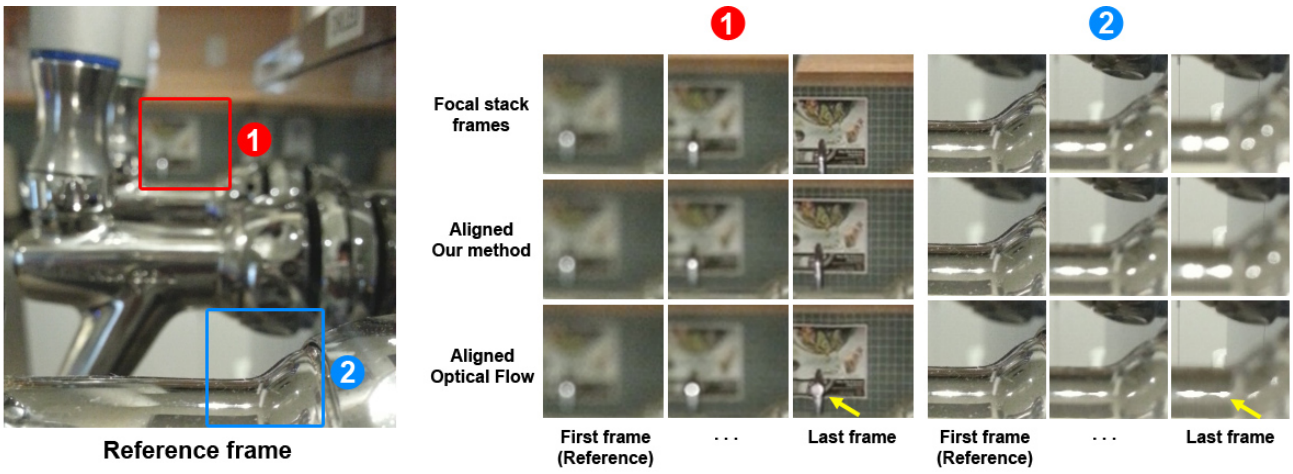

Figure 2. Alignment comparisons between our method and a standard optical flow algorithm with concatenation. Shown on the left, a reference frame in a focal stack that every frame has to align to. The top rows show the focal stack frames taken from each zoom-in box, the middle rows show our alignment results, and the bottom rows show results from a standard optical flow algorithm. In the red zoom-in, there is a downward translation in the focal stack frames that is corrected by both algorithms. However, the standard optical flow erroneously expands the white spot at the yellow arrow to resemble the bokeh in the reference. Similarly in the blue zoom-in, the bokehs in the last frame are erroneously contracted to match the in-focus highlights of the reference.

blur stack) that minimizes the intensity difference of a patch around the pixel. Specifically, we compute a difference map $\mathcal{D}_i : \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}$ by:

$$\mathcal{D}_i(x, y, r) = \sum_{(x',y')} w(x'-x, y'-y) \left| \hat{I}_i(x', y') - \hat{I}_0^r(x', y') \right|, \tag{7}$$

where w is a 2D Gaussian kernel centered at $(0, 0)$ with variance (μ^2, μ^2). For each frame in the focal stack $\hat{I}_i$ we compute a blur map $\mathcal{B}_i$ and an associated confidence map, $\mathcal{C}_i : \mathbb{R}^2 \to \mathbb{R}$ as

$$\mathcal{B}_i(x, y) = \delta_i \cdot \operatorname*{argmin}_r \mathcal{D}_i(x, y, r), \tag{8}$$

$$\mathcal{C}_i(x, y) \propto \left(\operatorname*{mean}_{r'} \mathcal{D}_i(x, y, r') - \min_{r'} \mathcal{D}_i(x, y, r') \right)^{\alpha}, \tag{9}$$

where δ_i is a scaling constant to undo the magnification compensation done in Section 4 and revert the radius back to its original size before alignment. This scale can be estimated from the same ICIA algorithm [2] by restricting the transformation to only scale.

Given $\mathcal{B}_i, \mathcal{C}_i$ for each frame, we jointly optimize for aperture size, focal depths, focal length, and a depth map by minimizing the following equation:

$$\min_{A,s,F,f_1,\ldots,f_n} \sum_{i=1}^{n} \sum_{x,y} \Big(\big(b_i(s_{xy}) - \mathcal{B}_i(x, y) \big) \cdot \mathcal{C}_i(x, y) \Big)^2. \tag{10}$$

This non-linear least squares problem can be solved using Levenberg-Marquardt. We initialize the focal depths with a linear function and the depthmap with the index map translated into the actual depth according to the initialized focal depths. The aperture and focal lengths are set arbitrarily to constants provided in Section 8. In our implementation, we use Ceres Solver [1] with sparse normal cholesky as the linear solver.

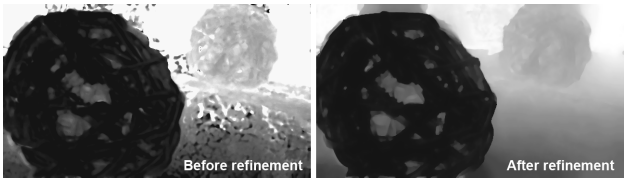

Figure 4. Depth maps before (left) and after (right) refinement.

6.2. Depth Map Refinement

The joint optimization gives good estimates for the aperture, focal length, and focal depths as the number of constraints is linear in the number of total pixels, which makes the problem partially over-constrained with respect to those global parameters. However, depth estimates are constrained by far fewer local neighborhood pixels and can be noisy as shown in Figure 4.

We therefore optimize for a refined depth map by fixing the aperture, focal length, and focal depths and reducing the problem to a better-behaved problem which is convex in the regularizer term. In particular, we use a global energy minimization framework where the data term is the photometric error $\mathcal{D}$ from the previous section, and employ an anisotropic regularization similar to [30] on the gradient of the inverse depth with the robust Huber norm. Around occlusion boundaries, the image-driven anisotropic regularizer decreases its penalty to allow for depth discontinuities,

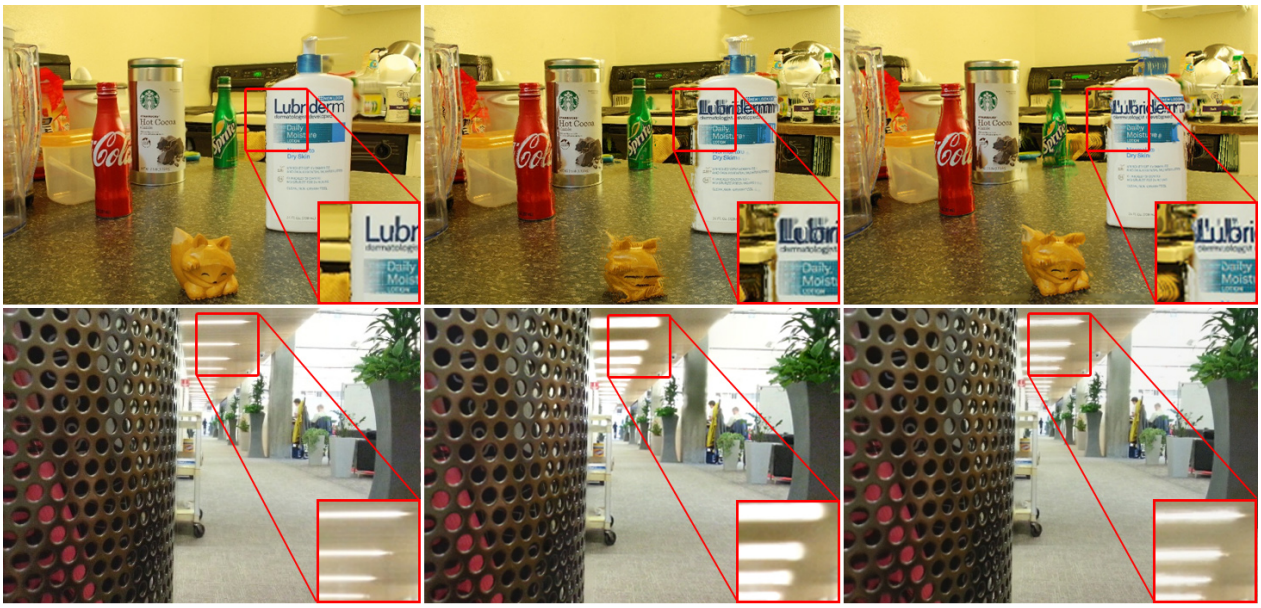

Figure 3. All-in-focus results produced from our algorithm (left), and two commercial applications: Photoshop CS5 (middle) and Helicon Focus (right). The first row shows bleeding artifacts due to bokehs in both commercial applications. The input focal stack for the second row contains a large translational motion and parallax. Photoshop and HeliconFocus fail to align the focal stack and produce substantially worse all-in-focus images compared to our method.

while promoting smooth depth maps elsewhere.

Let $\mathcal{Q}(\mathbf{x}) : \Omega \rightarrow \mathbb{R}$ be a functional that represents the inverse depth value at pixel $\mathbf{x} = (x, y)^\top$. The data term associated with each label at pixel $\mathbf{x}$ is computed as:

$$U(\mathbf{x}, \mathcal{Q}(\mathbf{x}))) = \frac{1}{n} \sum_{i}^{n} \mathcal{D}_i \left(\mathbf{x}, b_i(\mathcal{Q}(\mathbf{x}))\right), \qquad (11)$$

where n is the number of focal stack frames. The energy functional we seek to minimize is:

$$E_{\mathcal{Q}} = \int_{\Omega} \lambda U(\mathbf{x}, \mathcal{Q}(\mathbf{x})) + \|T(\mathbf{x}) \nabla \mathcal{Q}(\mathbf{x})\|_{\epsilon} \, \mathrm{d}\mathbf{x}, \qquad (12)$$

where $T(\mathbf{x})$ is a symmetric, positive definite diffusion tensor as suggested in [30] and defined as $\exp(-\alpha|\nabla I|^{\beta})\vec{n}\vec{n}^\top + \vec{n}^\perp\vec{n}^{\perp\top}$ where $\vec{n} = \frac{\nabla I}{|\nabla I|}$ and $\vec{n}^\perp$ is a unit vector perpedicular to $\vec{n}$, and $\|z\|_{\epsilon}$ is a Huber norm defined as.

$$\|z\|_{\epsilon} = \begin{cases} \frac{\|z\|_2^2}{2\epsilon} & \|z\|_2 \leq \epsilon \\ \|z\|_1 - \frac{\epsilon}{2} & \text{otherwise} \end{cases} \qquad (13)$$

Following the approach of [25, 20], the data term and smoothness term in (12) are decoupled through an auxiliary functional $\mathcal{A}(\mathbf{x}) : \Omega \rightarrow \mathbb{R}$ and split into two minimization problems which are alternately optimized:

$$E_{\mathcal{A}} = \int_{\Omega} \lambda U(\mathbf{x}, \mathcal{A}(\mathbf{x})) + \frac{1}{2\theta}(\mathcal{A}(\mathbf{x}) - \mathcal{Q}(\mathbf{x}))^2 \, \mathrm{d}\mathbf{x}, \quad (14)$$

$$E_{\mathcal{Q}} = \int_{\Omega} \frac{1}{2\theta}(\mathcal{A}(\mathbf{x}) - \mathcal{Q}(\mathbf{x}))^2 + \|T(x) \nabla \mathcal{Q}(\mathbf{x})\|_{\epsilon} \, \mathrm{d}\mathbf{x}. \quad (15)$$

As $\theta \rightarrow 0$, it can be shown [9] that this minimization is equivalent to minimizing equation 12. Equation 14 is minimized using a point-wise search over the depth labels which can handle fine structures without resorting to linearizing the data term or a coarse-to-fine approach such as [25]. Similar to [20], we perform a single Newton step around the minimum point to achieve sub-sample accuracy. Equation 15 is similar to the ROF image denoising problem [22] with an anisotropic Huber norm and is minimized using a primal-dual algorithm [10] through the Legendre-Fenchel transform.

7. Handling Bokeh

Defocusing highlights and other saturated regions create sharp circular expansions known as *bokeh*. This effect can cause artifacts if not properly accounted for during image alignment. Optical flow algorithms will explain the bokeh expansion as if it was caused by parallax and, after alignment, it will contract or expand the bokeh to match the reference frame, resulting in a physically incorrect aligned focal stack, and "bleeding" artifacts in the all-in-focus image, e.g., see top row in Figure 3.

To solve this problem, we propose a technique to detect bokeh regions by looking for bright areas and measure each pixel's expansion through the focal stack. The measure of

expansion is then used to regularize the optical flow in regions where bokeh is present.

7.1. Bokeh Detector

Bokeh occurs in bright regions of the image which expand through the focal stack. As a discrete approximation of how much each pixel in each frame expands, we propose a voting scheme where every pixel votes for the pixels that it flows to in every other frame. Pixels with most votes correspond to sources of expansion in the stack. Specifically, we first compute a low-regularized optical flow and use the concatenation technique of Section 4 to compute all-pair $\mathcal{F}_i^j$ for all $i, j \in [n]$.

Let $\mathbf{p}_i(u, v)$ be the pixel at (u, v) at frame i. $\mathbf{p}_i(u, v)$ will vote for pixels $\mathbf{p}_j(u+\mathcal{F}_i^j(u,v)_x, v+\mathcal{F}_i^j(u,v)_y)$ for all $j \neq i$. To avoid discretization artifacts, each vote is spread to pixels in a small neighborhood with weights according to a Gaussian falloff. Let $u' = u + \mathcal{F}_j^i(u,v)_x$ and $v' = v + \mathcal{F}_j^i(u,v)_y$, the total vote for pixel $\mathbf{p}_j(s,t)$ is computed as

$$\mathcal{V}_j(s,t) = \frac{1}{n-1} \sum_{i \neq j} \sum_{u,v} \exp\left(-\frac{(u'-s)^2 + (v'-t)^2}{2\sigma^2}\right) \tag{16}$$

For a given frame j in the focal stack, we then threshold $V_j > \tau_v$ and the color intensity $I_j(s,t) > \tau_I$ to detect the sources of bokeh expansion. To detect bokeh pixels in all the frames in the stack the maximum votes are propagated back to the corresponding pixels in every other frame and a bokeh confidence map for each frame is generated as

$$\mathcal{K}_i(s,t) = \max_{j \neq i} (\mathcal{W}_{\mathcal{F}_j^i}(\mathcal{V}_j))(s,t) \tag{17}$$

7.2. Bokeh-aware Focal Stack Alignment

Since optical flow estimates are inaccurate at bokeh boundaries, special care is needed to infer flow in these regions (e.g., by locally increasing flow regularization). In our implementation, we perform "flow inpainting" as a post-processing step, which makes correcting bokeh independent of the underlying optical flow algorithms used. For each $\mathcal{F}_i^n$, we mask out areas with high $\mathcal{K}_i$, denoted by Ω with boundary $\partial\Omega$, and interpolate the missing flow field values by solving Fourier's heat equation:

$$\min_{\mathcal{F}'_x, \mathcal{F}'_y} \iint_\Omega \|\nabla \mathcal{F}'_x\|^2 + \|\nabla \mathcal{F}'_y\|^2 \, \mathrm{d}u \, \mathrm{d}v \tag{18}$$

$$\text{s.t. } \mathcal{F}'_x|_{\partial\Omega} = \mathcal{F}_x|_{\partial\Omega}, \mathcal{F}'_y|_{\partial\Omega} = \mathcal{F}_y|_{\partial\Omega} \tag{19}$$

This can be turned into a linear least squares problem on discrete pixels by computing gradients of pixels using finite differences. The boundary condition can also be encoded as a least squares term in the optimization, which can be solved efficiently.

7.3. Bokeh-aware All-in-Focus Image Stitching

The previous flow interpolation step allows us to preserve the original shapes of the Bokehs in the focal stack. However, since bokehs have high-contrast contours in every frame, the sharpness measure used to stitch the all-in-focus image tends to select bokeh contours and therefore produces bleeding artifacts (Figure 3) around bokeh regions. To fix this, we incorporate the bokeh detector into a modified data term E' of the previous MRF formulation as follows:

$$E'_i(x_i = j) = \begin{cases} \alpha E_i(x_i) + (1-\alpha) C_i(x_i) & \text{if } \mathcal{K}_i(s,t) > 0 \\ E_i(x_i) & \text{otherwise} \end{cases} \tag{20}$$

where $E_i(x_i)$ is the original data term, $C_i(x_i) = I_i(s,t)$ is a color intensity term so that larger bokehs are greater penalized, and α is a balancing constant.

8. Experiments

We now describe implementation details, runtime, results, evaluations, and applications.

Implementation details Flows in Section 4 are computed using Ce Liu's [16] implementation of optical flow (based on Brox et al.[7] and Bruhn et al. [8]) with parameters (α, ratio, minWidth, outer-,inner-,SOR-iterations) = $(0.03, 0.85, 20, 4, 1, 40)$. Flows in Section 7.1 are computed using the same implementation with α = 0.01. In Section 5, each color channel is scaled to 0-1 range, the weight $\lambda = 0.04$, and $\mu = 13$. In Section 6.1, we blur the all-in-focus using a radius starting from 0 up to 6.5 pixels by a 0.25 increment. The exponential constant $\alpha = 2$, and $\mu = 15$. For Levenberg-Marquardt, the nearest and farthest depths are set to 10 and 32. The focal depth and aperture are set to 2 and 3. In the refinement step 6.2, we quantize the inverse depth into 32 bins lying between the minimum and maximum estimated depths from the calibration step. The balancing term $\lambda = 0.001$. The tensor constants are $\alpha = 20, \beta = 1$, and the Huber constant $\epsilon = 0.005$. The decoupling constant starts at $\theta^0 = 2$ and $\theta^{n+1} = \theta^n(1 - 0.01n), n \leftarrow n+1$ until $\theta \leq 0.005$. The Newton step is computed from the first-order and second-order central finite difference of the data term plus the quadratic decoupling term. In Section 7.1 equation 16, the standard deviation $\sigma = 3$, and thresholds $\tau_v = 5, \tau_I = 0.5$. In Section 7.3, the balancing term $\alpha = 0.7$.

Runtime We evaluate runtime on the "balls" dataset (Figure 7) with 25 frames at 640x360 pixels on a single CPU core of Intel i7-4770@3.40 Ghz. The complete pipeline takes about 20 minutes which includes computing optical flows (8 minutes), detecting bokehs (48 seconds), focal stack alignment (45 seconds), bokeh-aware all-in-focus stitching (14 seconds), focal stack calibration (8 minutes), and depth map refinement (3 minutes). The ma-

Figure 5. Multiple datasets captured with a hand-held Samsung Galaxy phone. From left to right (number of frames in parenthesis): plants(23), bottles(31), fruits(30), metals(33), window(27), telephone(33). Top row shows the all-in-focus stitch. Bottom row shows the reconstructed depth maps.

jority (75%) of the runtime is spent on computing optical flow and rendering the focal stack which are part of the calibration and refinement steps. We believe these costs can be brought down substantially with an optimized, real-time optical flow implementation, e.g., [32] which reduces the optical flow runtime to 36 seconds.

Experiments We present depth map results and all-in-focus images in Figure 5 for the following focal stack datasets (number of frames in parenthesis): plants(23), bottles(31), fruits(30), metals(33), window(27), telephone(33). For each dataset, we continuously captured images of size 640x360 pixels using a Samsung Galaxy S3 phone during auto-focusing. The results validate the proposed algorithm and the range of scenes in which it works, e.g., on specular surfaces in fruits and metals. The window dataset shows a particularly interesting result. The depthmap succesfully captures the correct depth for rain drops on the window.

We demonstrate our aligned focal stack results in the supplementary video and through a comparison of all-in-focus photos generated by our method, Adobe Photoshop CS5, and HeliconFocus in Figure 3. We use the same sequences as input to these programs. The kitchen sequence (Figure 3 top) was captured with a Nikon D80 at focal length 22mm by taking multiple shots while sweeping the focal depth. The motion of the camera contains large translation and some rotation. Photoshop and HeliconFocus fail to align the focal stack frames and produce alignment artifacts in the all-in-focus photos as shown in the zoom-in boxes whereas our method produces much fewer artifacts. The bottom row shows all-in-focus results from a sequence captured with a Samsung Galaxy S3 phone, hand-held camera motion and almost no lateral parallax. The motion in this sequence is dominated by the magnification change, which is a global transformation, and all three techniques can align the frames equally well. However, our method is able to produce an all-in-focus that does not suffer from Bokeh bleeding, for example on the ceiling lights while Photoshop and HeliconFocus show the bleeding problem on the ceiling as well as near the concrete column.

Evaluations We evaluated our algorithm on 14-frame focal stack sequences with no, small, and large camera motions. We captured these sequences using a Nikon D80 with a 18-135mm lens at 22mm (our mobile phone does not allow manual control on focus to generate such ground-truth). The scene consists of 3 objects placed on a table: a box of playing cards at 12 inches from the camera's center of projection, a bicycling book at 18.5 inches, and a cook book at 28 inches. The background is at 51 inches. The first sequence contains no camera motion, and only the magnification change caused by lens breathing during focusing. The closest focus is at the box of playing cards and the furthest is at the background. The second sequence has a small 0.25-

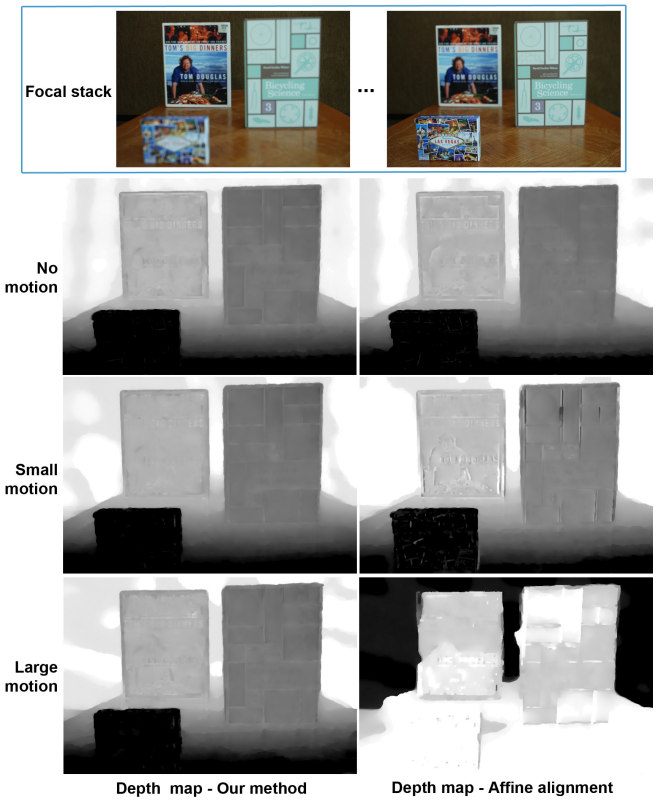

Figure 6. Our evaluation focal stack is shown on top. Next rows show depth maps from our alignment algorithm vs affine alignment algorithm in three different scenarios: a static scene, a scene with a small (0.25-inches) and large (1-inch) camera motion. Depth map estimation using affine alignment produces higher errors and more artifacts around areas with strong image gradients. The calibration completely fails in the large motion case.

Table 1. Results from our method.

Motion:	none	small	large
Bike book (18.5)	16.94	16.57	16.71
Cook book (28)	24.58	22.82	22.85
RMS Error (inches)	2.66	3.91	3.86

inch translational motion generated by moving the camera on a tripod from left to right between each sweep shot. The third sequence has a large 1-inch translational motion generated similarly. Since our depth reconstruction is up to an affine ambiguity in inverse depth, we cannot quantify an absolute metric error. Instead, we solve for 2-DoF affine parameters α and β in $\frac{1}{\hat{s}} = \frac{\alpha}{s} + \beta$ such that they fit the depth of the box of playing cards s_{box} and the background s_{bg} to the ground-truth depth values $\hat{s}_{\text{box}}$ and $\hat{s}_{\text{bg}}$. The depths of the two books averaged over each surface are reported in table 1 and the corresponding depth maps are shown in Figure 6.

We also compare depth maps from our algorithm to a representative of previous work by replacing optical flow alignment with affine alignment (Figure 6). We apply the same concatenation scheme to affine alignment to handle the defocus variation. Results show that affine alignment cannot handle even small amounts of scene parallax. It produces many depth artifacts in the small-motion sequence and completely fails to estimate reasonable focal depths in the large-motion sequence. Errors are shown in table 2. For evaluations of the calibration process, please see our supplementary materials.

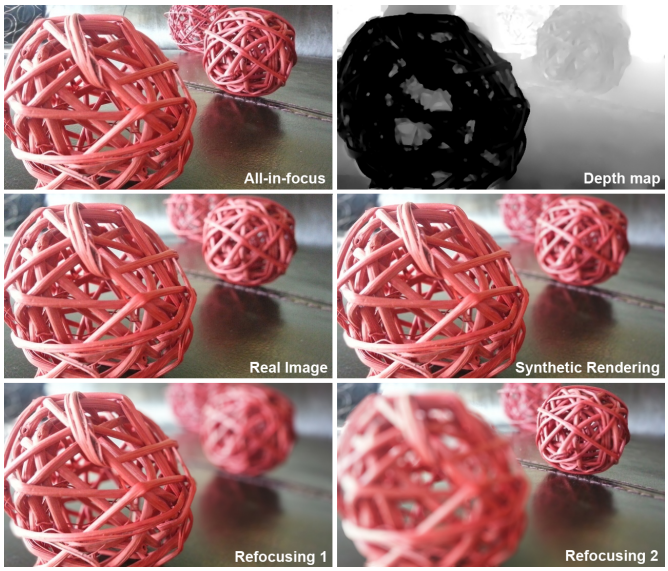

Figure 7. Given a recovered all-in-focus (top left) and a depth map (top right), we can synthetically render any frame in the focal stack and compare to qualitatively verify the calibration process. The middle right image is a synthetic rendering of scene by bluring the all-in-focus according to the estimated depth and camera parameters to match the real image in the middle left. The bottom left and right images show a refocusing application which simulates a larger aperture to decrease the depth-of-field and changes of focus.

Table 2. Results from affine alignment.

Motion:	none	small	large
Bike book (18.5)	16.92	17.63	Failed
Cook book (28)	24.43	50.96	Failed
RMS Error (inches)	2.76	16.25	Failed

Application The reconstructed depthmap enables interesting rerendering capabilities such as increasing the aperture size to amplify the depth-of-field effect as shown in Figure 7, or extend the focus beyond the recorded set, and synthesize a small-baseline perspective shift.

9. Conclusion

We introduced the first depth from focus (DfF) method capable of handling images from mobile phones and other hand-held cameras. We formulated a novel uncalibrated DfD problem and proposed a new focal stack aligning algorithm to account for scene parallax. Our approach has been demonstrated on a range of challenging cases and produces high quality results.

References

[1] Sameer Agarwal, Keir Mierle, and Others. Ceres solver. `https://code.google.com/p/ceres-solver/`.

[2] Simon Baker and Iain Matthews. Equivalence and efficiency of image alignment algorithms. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–1090. IEEE, 2001.

[3] Rami Ben-Ari. A unified approach for registration and depth in depth from defocus. 2014.

[4] V Michael Bove Jr et al. Entropy-based depth from focus. *JOSA A*, 10(4):561–566, 1993.

[5] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(9):1124–1137, 2004.

[6] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(11):1222–1239, 2001.

[7] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In *Computer Vision-ECCV 2004*, pages 25–36. Springer, 2004.

[8] Andrés Bruhn, Joachim Weickert, and Christoph Schnörr. Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. *International Journal of Computer Vision*, 61(3):211–231, 2005.

[9] Antonin Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical imaging and vision*, 20(1-2):89–97, 2004.

[10] Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.

[11] Trevor Darrell and Kwangyoen Wohn. Pyramid based depth from focus. In *Computer Vision and Pattern Recognition, 1988. Proceedings CVPR'88., Computer Society Conference on*, pages 504–509. IEEE, 1988.

[12] John Ens and Peter Lawrence. A matrix based method for determining depth from focus. In *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*, pages 600–606. IEEE, 1991.

[13] Ray A Jarvis. A perspective on range finding techniques for computer vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (2):122–139, 1983.

[14] Vladimir Kolmogorov and Ramin Zabin. What energy functions can be minimized via graph cuts? *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(2):147–159, 2004.

[15] Eric Krotkov. Focusing. *International Journal of Computer Vision*, 1(3):223–237, 1988.

[16] Ce Liu. *Beyond pixels: exploring new representations and applications for motion analysis*. PhD thesis, Citeseer, 2009.

[17] Yifei Lou, Paolo Favaro, Andrea L Bertozzi, and Stefano Soatto. Autocalibration and uncalibrated reconstruction of shape from defocus. In *CVPR*, 2007.

[18] Aamir Saeed Malik and Tae-Sun Choi. A novel algorithm for estimation of depth map using image focus for 3d shape recovery in the presence of noise. *Pattern Recognition*, 41(7):2200–2225, 2008.

[19] Shree K Nayar and Yasuo Nakagawa. Shape from focus. *Pattern analysis and machine intelligence, IEEE Transactions on*, 16(8):824–831, 1994.

[20] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2320–2327. IEEE, 2011.

[21] Alex Paul Pentland. A new sense for depth of field. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (4):523–531, 1987.

[22] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variasion based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268, 1992.

[23] Rajiv Ranjan Sahay and Ambasamudram N Rajagopalan. Dealing with parallax in shape-from-focus. *Image Processing, IEEE Transactions on*, 20(2):558–569, 2011.

[24] Nitesh Shroff, Ashok Veeraraghavan, Yuichi Taguchi, Oncel Tuzel, Amit Agrawal, and R Chellappa. Variable focus video: Reconstructing depth and video for dynamic scenes. In *Computational Photography (ICCP), 2012 IEEE International Conference on*, pages 1–9. IEEE, 2012.

[25] Frank Steinbrucker, Thomas Pock, and Daniel Cremers. Large displacement optical flow computation withoutwarping. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1609–1614. IEEE, 2009.

[26] Gopal Surya and Murali Subbarao. Depth from defocus by changing camera aperture: A spatial domain approach. In *Computer Vision and Pattern Recognition, 1993. Proceedings CVPR'93., 1993 IEEE Computer Society Conference on*, pages 61–67. IEEE, 1993.

[27] Jay Martin Tenenbaum. Accommodation in computer vision. Technical report, DTIC Document, 1970.

[28] Masahiro Watanabe and Shree K Nayar. Telecentric optics for computational vision. In *Computer VisionECCV'96*, pages 439–451. Springer, 1996.

[29] Masahiro Watanabe and Shree K Nayar. Telecentric optics for focus analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(12):1360–1365, 1997.

[30] Manuel Werlberger, Werner Trobin, Thomas Pock, Andreas Wedel, Daniel Cremers, and Horst Bischof. Anisotropic huber-l1 optical flow. In *BMVC*, volume 1, page 3, 2009.

[31] Yalin Xiong and Steven A Shafer. Moment and hypergeometric filters for high precision computation of focus, stereo and optical flow. *International Journal of Computer Vision*, 22(1):25–59, 1997.

[32] Christopher Zach, Thomas Pock, and Horst Bischof. A duality based approach for realtime tv-l 1 optical flow. In *Pattern Recognition*, pages 214–223. Springer Berlin Heidelberg, 2007.

[33] Quanbing Zhang and Yanyan Gong. A novel technique of image-based camera calibration in depth-from-defocus. In *Intelligent Networks and Intelligent Systems, 2008. ICINIS'08. First International Conference on*, pages 483–486. IEEE, 2008.

[34] Changyin Zhou, Daniel Miau, and Shree K Nayar. Focal sweep camera for space-time refocusing. 2012.