Depth from Focus with Mobile Phone

Xinyu Liu
xinyuli2@andrew.cmu.edu
Robotics Institution

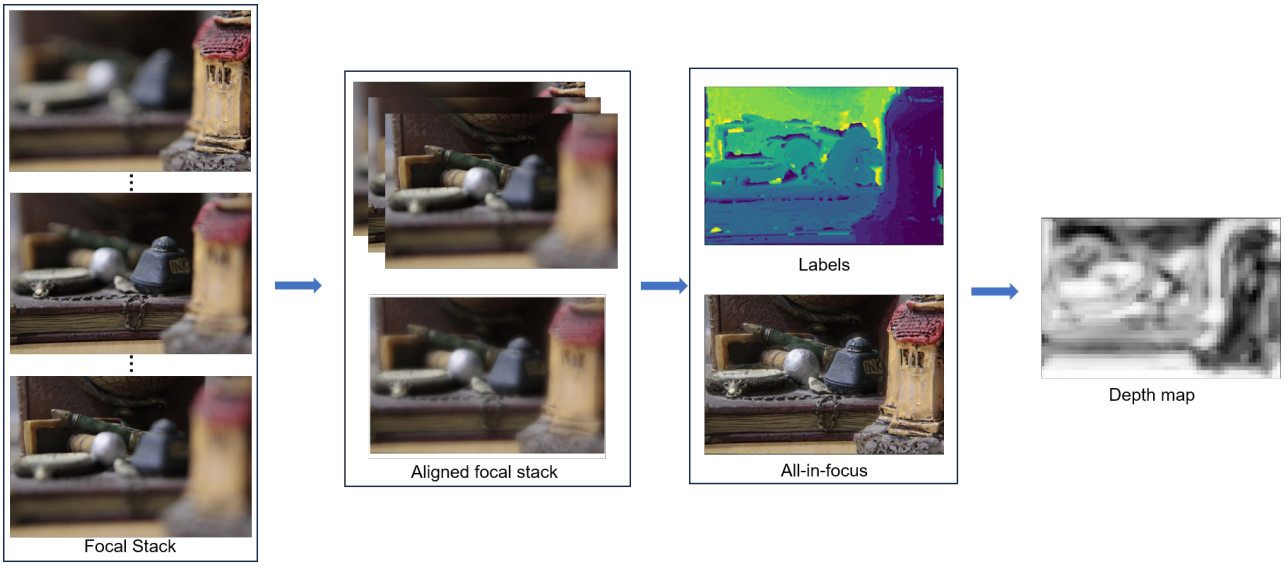


Figure 1: Our project's pipeline. From the focal stack captured with a mobile device, we first computed an aligned focal stack using the Inverse Compositional Alignment algorithm. Using this aligned dataset, we conduct Markov Random Field (MRF) optimization to generate an all-in-focus image. Finally, by leveraging both the aligned focal stack and the all-in-focus image, we formulate and solve a non-linear optimization problem to jointly optimize the depth map and the camera parameters.

ABSTRACT

While most depth from focus and defocus techniques operated on laboratory scenes, [15] introduced 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 project aims to implement the algorithms proposed in [15] so that we can produce an all-in-focus image and depth map from the focal stack captured using handheld devices.

KEYWORDS

Depth-from-focus, Mobile handsets, Optimization, Cameras

ACM Reference Format:
Xinyu Liu. 2023. Depth from Focus with Mobile Phone. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Although depth-from-focus (DfF) techniques have been researched for many years, they have mostly been limited to laboratory settings. To create a focal stack, one needs a tripod and a camera with interchangeable lenses, as well as a lens that allows for manual focusing.

Working on hand-held devices, such as standard mobile phones, can be more challenging for two reasons. Firstly, while almost all Depth from Focus (DfF) methods require calibrated capture, supporting commodity mobile phones requires working in an uncalibrated setting. Secondly, since human hands cannot be fully stable when capturing images, a focal sweep captured with a hand-held camera inevitably produces motion parallax. The general DfF methods used special optics (e.g. [16] , or employed simple global transformations to align images. [14] attempts to handle dynamic scenes, which can exhibit parallax, but this method still requires a calibrated camera in the lab. [15], which we aimed to re-implement in our project, proposed an alignment technique based on flow concatenation. We used a similar but simpler approach and successfully demonstrated uncalibrated DfF on hand-held devices.

We address the hand-held DfF problem in three steps: 1) focal stack alignment, 2) all-in-focus image generation, and 3) 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 [16]. In the second step, given an aligned focal stack, we aim to recover the all-in-focus image. To solve this problem, we used an MRF-based approach and solved the MRF energy optimization problem using graph cut. In the final step, we recover the depth map by formulating a non-linear optimization problem that jointly solves for camera settings and scene depth, which best explains the focal stack.

2 RELATED WORK

Calibrated DfF. Prior work exists for geometrically aligning frames in a calibrated image sequence: [4, 18] used an image warping approach to correct for magnification change. [14] proposed 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. [14] 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.

Uncalibrated DfF. [12] 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. [17] considered a related uncalibrated defocus problem for the special case where only the aperture changes between two images. [15] addressed the case where the aperture and focal length are unknown and formulated an optimization problem to jointly solve for camera settings, focal depths and the depth map.

Deep Learning Methods. With Convolutional Neural Networks (CNNs) becoming more popular in recent years, CNNs have been employed to learn an implicit relation between color pixels and depth [11, 5]. [10] proposed a fully convolutional architecture to model the ambiguous mapping between monocular images and depth maps. [6] proposed a deep depth from focus network which uses an auto-encoder-style convolutional neural network that outputs a disparity map from a focal stack.

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).

4 FOCAL STACK ALIGNMENT

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

Other work corrected for magnification changes through scaling and translating [4] or a similarity transform [14]. However, these global transformations are inadequate for correcting local parallax. Instead, [15] proposed 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. This problem is overcome by concatenating flows between consecutive frames in the focal stack, which ensures that the defocus between two input frames to the optical flow appears similar.

Given a set of frames in a focal stack $I_1, I_2, ..., I_n$, we assume without loss of generality that I_1 is the reference frame, which has the largest magnification. Our task is to align $I_2, ..., I_n$ 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 $\mathcal{W}_{\mathbf{F}}(I)$ denote the warp of image I according to the flow $\mathcal{F}$:

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

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 can then compute the flow between consecutive frames $\mathcal{F}_2^1, \mathcal{F}_3^2, ..., \mathcal{F}_n^{n-1}$, 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, $\mathcal{F}_x$ and $\mathcal{F}_y$ are images and are warped according to flow $\mathcal{F}'$. After this step, we can produce an aligned frame $\hat{I}_i = \mathcal{W}_{\mathcal{F}_i^1}(I_i)$.

However, the above case is under the ideal condition, where the difference between frames can be modeled by a simple translation. In reality, the flow between frames can be more complicated. Instead, we computed an affine warp using the Inverse Compositional Image Alignment algorithm described in [1]. To warp image I_{i+1} to image I_i, we need to minimize the image alignment objective function (2):

$$\sum_{(u,v)} \left(I_{i+1}(\mathcal{W}_{\mathcal{F}_{i+1}^i}(u,v)) - I_i(u,v) \right)^2 \tag{2}$$

5 ALL-IN-FOCUS IMAGE STITCHING

Given an aligned focal stack $\hat{I}_1, \hat{I}_2, ..., \hat{I}_n$, 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 [7, 9, 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:

$$E(x) = \sum_{i \in \mathcal{V}} E_i(x_i) + \lambda \sum_{(i,j) \in \mathcal{E}} E_{ij}(x_i, x_j) \tag{3}$$

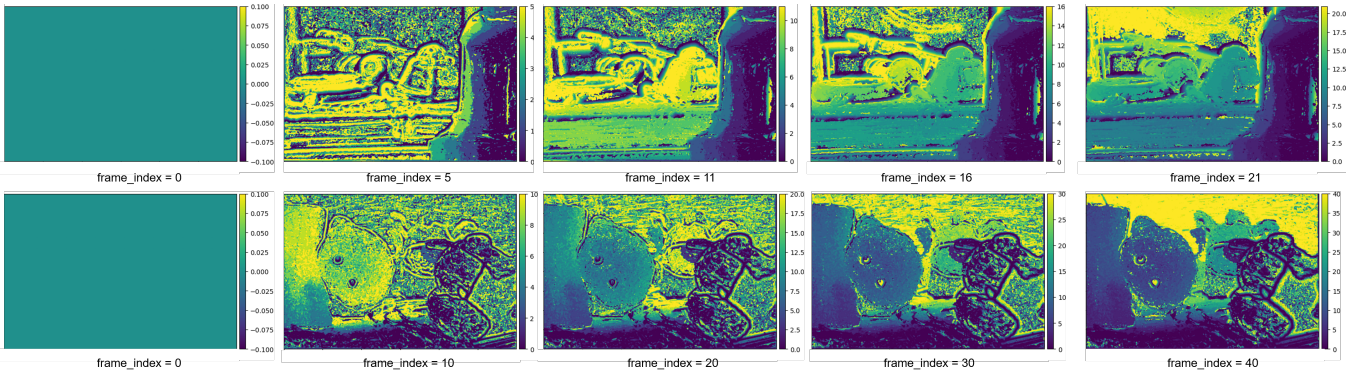


Figure 2: Intermediate results of all-in-focus image stitching through MRF optimization. The first row displays the outcome for dataset 05_castle, and the second row presents the result for dataset 07_toys. The weighting constant λ is configured to be 0.

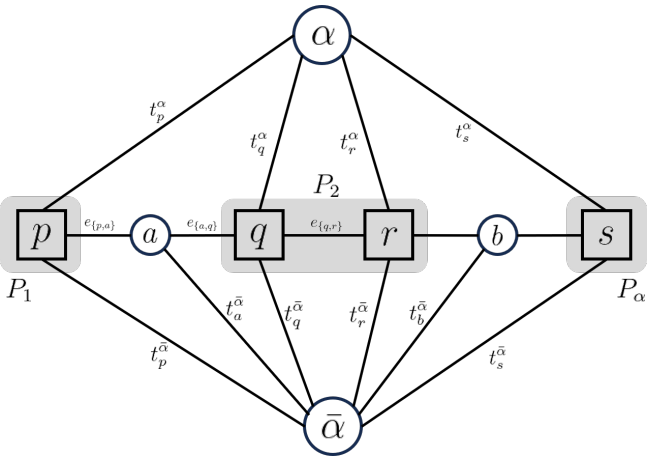


Figure 3: An illustration of the graph $\mathcal{G}$. α and $\bar{\alpha}$ are the terminal nodes. Every pixel in the image corresponds to a node, and adjacent pixels are represented as neighboring nodes. Nodes with identical labels, such as node q and node r, are directly connected by an edge. In the case of neighboring nodes with different labels, such as p and q, an auxiliary node and auxiliary edges are introduced. The edge weights are determined through computation involving the unary and pairwise terms

where x_i and x_j denote the i-th and the j-th frame, and 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 (σ^2, σ^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 $(|x_i - x_j|)$.

This MRF minimization problem can be solved using graph cut [3, 2, 8]. It is easy to prove that the pairwise term $E_{ij}(x_i, x_j)$ is submodular, namely, it satisfies the three constraints:

$$\begin{aligned} &E_{ij}(x_i, x_j) = 0 \Leftrightarrow x_i = x_j \\ &E_{ij}(x_i, x_j) = E_{ji}(x_j, x_i) \\ &E_{ij}(x_i, x_j) \leq E_{ik}(x_i, x_k) + E_{kj}(x_k, x_j) \end{aligned} \tag{4}$$

Therefore, the problem can be minimized using the α-expansion algorithm [3], which we briefly described in Algorithm 1.

Algorithm 1 α-expansion

```
Start with an arbitrary labeling f
Set success ← 0
while True do
  Set success ← 0
  for each label α ∈ ℒ do
    Find f̂ = arg min E(f′) among f′ within one α-expansion of f
    if E(f̂) < E(f) then
      Set f ← f̂ and success ← 1
    end if
  end for
  if success = 0 then
    break
  end if
end while
return f
```

Given an input labeling f and a label α, we want to find a labeling $\hat{f}$ that minimizes E over all labelings within one α-expansion of f. This is going to be done by computing a labeling corresponding to a minimum cut on a graph $\mathcal{G}_\alpha = (\mathcal{V}_\alpha, \mathcal{E}_\alpha)$. The structure of this graph is dynamically determined by the current labeling f and by the label α. The constructed graph is illustrated in Fig 3. The set of vertices includes the two terminals α and $\bar{\alpha}$, as well as all image pixels $p \in \mathcal{P}$. Additionally, for each pair of neighboring pixels p, q such that $f_p \neq f_q$, we create an auxiliary node $a_{p,q}$. Each pixel p is connected to the terminals α and $\bar{\alpha}$, which are called t-links. Each set of pixels p, q which are neighbors and $f_p = f_q$ are connected with an n-link. For each pair of neighboring pixels such that $f_p \neq f_q$, we create a triplet of edges $\{e_{p,a}, e_{a,q}, t_a^\alpha\}$. The set of edges is then

$$\mathcal{E} = \left\{ \bigcup_{p \in \mathcal{P}} \{t_p^\alpha, t_p^{\bar{\alpha}}\}, \bigcup_{\{p,q \in \mathcal{N}, f_p \neq f_q\}} \mathcal{E}_{\{p,q\}}, \bigcup_{\{p,q \in \mathcal{N}, f_p = f_q\}} e_{\{p,q\}} \right\} \tag{5}$$

The weights of the edges are defined in Table 1.

edge	weight	for
$t_p^{\bar{\alpha}}$	∞	$p \in \mathcal{P}_\alpha$
$t_p^{\bar{\alpha}}$	$E_p(x_p)$	$p \notin \mathcal{P}_\alpha$
t_p^{α}	$E_p(\alpha)$	$p \in \mathcal{P}$
$e_{\{p,a\}}$	$E(x_p, \alpha)$	$\{p, q \in \mathcal{N},\ f_p \neq f_q\}$
$e_{\{a,q\}}$	$E(\alpha, x_q)$	
t_a^{α}	$E(x_p, x_q)$	
$e_{\{p,q\}}$	$E(x_p, \alpha)$	$\{p, q\} \in \mathcal{N},\ f_p = f_q$

Table 1: The weights of the edges.

We then solve for the minimum cut of the graph, with α and $\bar{\alpha}$ as the source and destination vertices. There is a one-to-one correspondence between a cut and a labeling, demonstrated in (6)

$$f_p^C = \begin{cases} \alpha & \text{if} \quad t_p^{\alpha} \in C \\ f_p & \text{if} \quad t_p^{\bar{\alpha}} \in C \end{cases} \quad \forall p \in \mathcal{P} \tag{6}$$

For edges in the minimum cut thas has α as one of its vertices, we update the pixel represented by the other node to α.

6 FOCAL STACK CALIBRATION AND DEPTH MAP RECONSTRUCTION

Given an aligned focal stack $\hat{I}_1, \hat{I}_2, ..., \hat{I}_n$, 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 $f_1, ..., f_n$ and a depth map representing the scene $s : \mathbb{R}^2 \to [0, \infty)$. Assuming the scene is Lambertian and is captured by a camera following a thin-lens model, and the a uniform disc-shaped point spread function (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:

$$\hat{I}_i = \iint r_{uv} D(x - u, y - v, b_i(s_{uv}))du\ dv \tag{7}$$

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:

$$b_i(s) = A \cdot \frac{|f_i - s|}{s} \cdot \frac{F}{f_i - F} \tag{8}$$

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

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 δ_r 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 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:

$$\mathcal{D}_i(x, y, r) = \sum_{(x', y')} w(x' - x, y' - y) \left| \hat{I}_i(x', y') - \hat{I}_0^r(x', y') \right| \tag{9}$$

,

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, $C_i : \mathbb{R}^2 \to \mathbb{R}$ as

$$\mathcal{B}_i(x, y) = \sigma_i \cdot argmin \mathcal{D}_i(x, y, r) \tag{10}$$

$$C_i(x, y) = \left(mean\ \mathcal{D}_i(x, y, r') - min\ \mathcal{D}_i(x, y, r') \right)^{\alpha} \tag{11}$$

where σ_i is a scaling constant.

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

$$min \sum_{i=1}^{n} \sum_{x,y} \left(\left(b_i(s_{xy}) - \mathcal{B}_i(x, y) \right) \cdot C_i(x, y) \right)^2 \tag{12}$$

We solved this non-linear least squares problem using . We initialize the focal depths with a linear function, and the depth map as an image with every pixel initialized to one. The aperture and focal lengths are set arbitrarily to constants provided in Section 7. We use the scipy's least_squares solver in our implementation.

7 EXPERIMENTS

We now describe implementation details, results, and applications.

Implementation details Focal stack alignment in Section 4 is computed using the Matthew-Bakers Inverse Compositional Alignment algorithm [1]. The stop condition threshold $\delta_{th} = 1e - 4$, and the maximum number of iterations is set to 100 for all focal stacks. For all-in-focus image stitching in Section 5, the variance of the Gaussian patch $\sigma = 1$, and the weight of the pairwise term $\lambda = 0.001$. We performed a qualitative study on the value of λ and the results are shown in Section 7.2.

For focal stack calibration in Section 6, the exponential constant $\alpha = 2$, the variance of Gaussian kernel $\sigma = 1$. The nearest and farthest depths are set to 10 and 38. The initial focal length and aperture are set to 2 and 3. We created different blur stacks with $\delta_r = 0.01, 0.05, 0.1$ and 0.2, the results are shown in Section 7.3.

Experiments We present all-in-focus images and the depth maps for the following focal stack datasets (number of frames in parenthesis): 05_castle(28), 07_toys(53), and mobo2_chip(47). For each dataset, we first reshaped the images from 1920×1080 to 960×640. Then we run focal stack alignment to remove the effect of camera movements. The resulting images are then resized to size 600×400. We used the resized images to generate the all-in-focus image stitching result using MRF optimization. Finally, the aligned image stack and the all-in-focus image are used to jointly optimize for the depth map, the focal length, the aperture size, and the focal depths.

Application The reconstructed depth map enables interesting rerendering capabilities such as increasing the aperture size to amplify the depth-of-field effect.

7.1 Focal Stack Alignment

The results of focal stack alignment are illustrated in Figure 5, where we have superimposed the frames from the original focal stack with those from the aligned focal stack for comparison. In

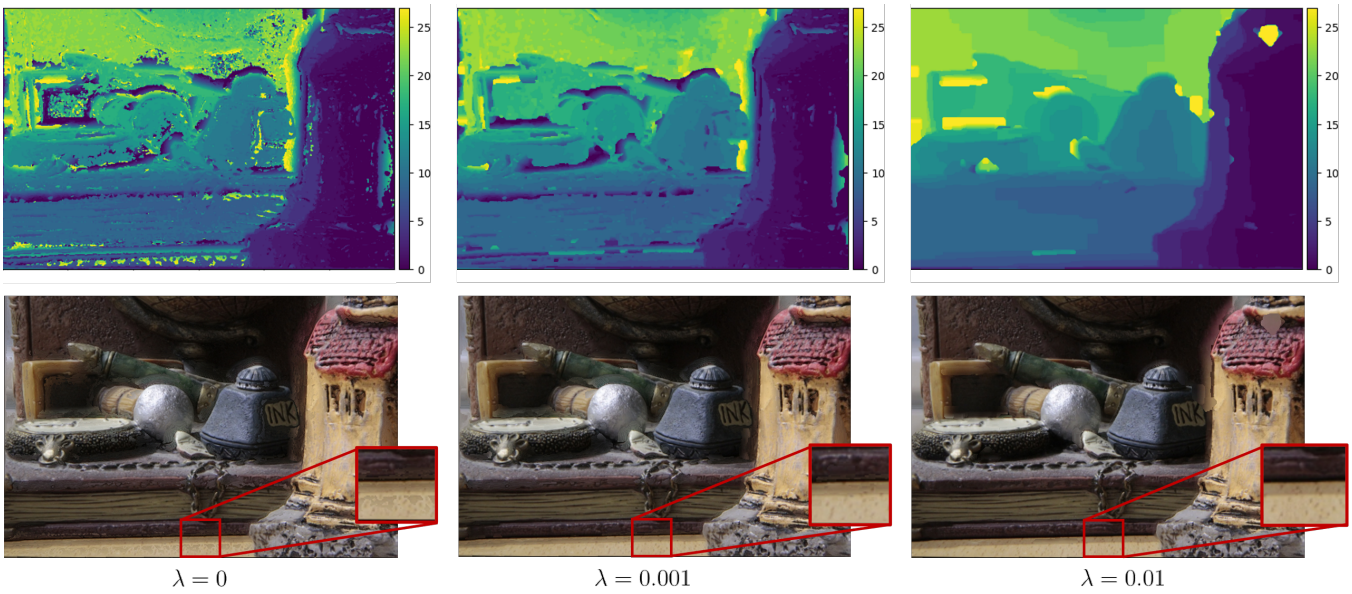


Figure 4: All-in-focus images reconstructed using different weighting constant λ. The first row illustrates the label image, where colors denote the frame index. The second row presents the reconstructed all-in-focus image. In the highlighted zoom-in region, for $\lambda = 0$, the reconstruction may capture more details, but it also introduces artifacts. With $\lambda = 0.01$, the reconstruction is smoother but may miss certain shape details. Additionally, it is noteworthy that in regions with low light intensity, the label image may yield incorrect results.

the case of datasets 05_castle and 07_toys, where the camera movement is primarily characterized by translation and is not notably pronounced, the benefits derived from the focal stack alignment are marginal. For the mobo_chip dataset, where the captured focal stack exhibits substantial motion parallax, the alignment step substantially enhances the overall result.

7.2 All-in-focus Image Stitching

We performed a qualitative study on the value of the weighting constant λ which determines how smooth the final result is. We experimented with $\lambda = 0.0, 0.001$ and 0.01. The label images and the final reconstructed all-in-focus images are shown in Figure 4.

7.3 Depth Map Reconstruction

We used the scipy package to solve the non-linear optimization problem.

8 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 in a range of challenging cases and produces reasonable results.

REFERENCES

[1] S. Baker and I. Matthews. "Equivalence and efficiency of image alignment algorithms". In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001.* Vol. 1. 2001, pp. I–I. DOI: 10.1109/CVPR.2001.990652.

[2] Y. Boykov and V. Kolmogorov. "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26.9 (2004), pp. 1124–1137. DOI: 10.1109/TPAMI.2004.60.

[3] Y. Boykov, O. Veksler, and R. Zabih. "Fast approximate energy minimization via graph cuts". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23.11 (2001), pp. 1222–1239. DOI: 10.1109/34.969114.

[4] T. Darrell and K. Wohn. "Pyramid based depth from focus". In: *Proceedings CVPR '88: The Computer Society Conference on Computer Vision and Pattern Recognition.* 1988, pp. 504–509. DOI: 10.1109/CVPR.1988.196282.

[5] David Eigen, Christian Puhrsch, and Rob Fergus. "Depth Map Prediction from a Single Image Using a Multi-Scale Deep Network". In: *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2.* NIPS'14. Montreal, Canada: MIT Press, 2014, pp. 2366–2374.

[6] Caner Hazirbas et al. *Deep Depth From Focus.* 2018. arXiv: 1704.01085 [cs.CV].

[7] R. A. Jarvis. "A Perspective on Range Finding Techniques for Computer Vision". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-5.2 (1983), pp. 122–139. DOI: 10.1109/TPAMI.1983.4767365.

[8] V. Kolmogorov and R. Zabin. "What energy functions can be minimized via graph cuts?" In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26.2 (2004), pp. 147–159. DOI: 10.1109/TPAMI.2004.1262177.

[9] Eric Krotkov. "Focusing". In: *International Journal of Computer Vision* 1 (2004), pp. 223–237. URL: https://api.semanticscholar.org/CorpusID:212694615.

[10] Iro Laina et al. "Deeper depth prediction with fully convolutional residual networks". In: *2016 Fourth international conference on 3D vision (3DV).* IEEE. 2016, pp. 239–248.

[11] Fayao Liu, Chunhua Shen, and Guosheng Lin. "Deep convolutional neural fields for depth estimation from a single image". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2014), pp. 5162–5170. URL: https://api.semanticscholar.org/CorpusID:13153.

[12] Yifei Lou et al. "Autocalibration and Uncalibrated Reconstruction of Shape from Defocus". In: *2007 IEEE Conference on Computer Vision and Pattern Recognition.* 2007, pp. 1–8. DOI: 10.1109/CVPR.2007.383210.

[13] 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". In: *Pattern Recogn.* 41.7 (July 2008), pp. 2200–2225. ISSN: 0031-3203. DOI: 10.1016/j.patcog.2007.12.014. URL: https://doi.org/10.1016/j.patcog.2007.12.014.

[14] Nitesh Shroff et al. "Variable focus video: Reconstructing depth and video for dynamic scenes". In: *2012 IEEE International Conference on Computational Photography (ICCP).* 2012, pp. 1–9. DOI: 10.1109/ICCPhot.2012.6215219.

Figure 6: Comparison between the frame in the focal stack and the all-in-focus image. The first row displays the result from the 05_castle dataset, the second row displays the result from the 07_toys dataset, and the third row displays the result from the mobo_chip dataset.

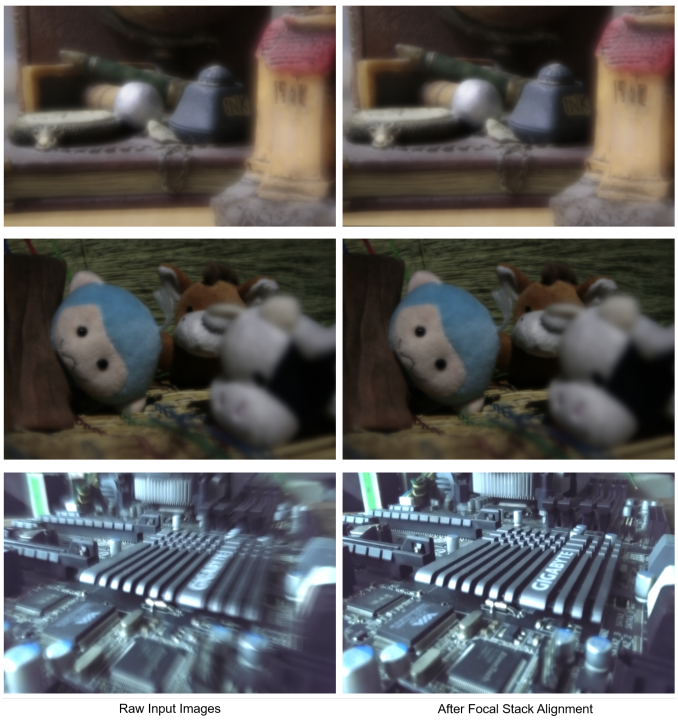


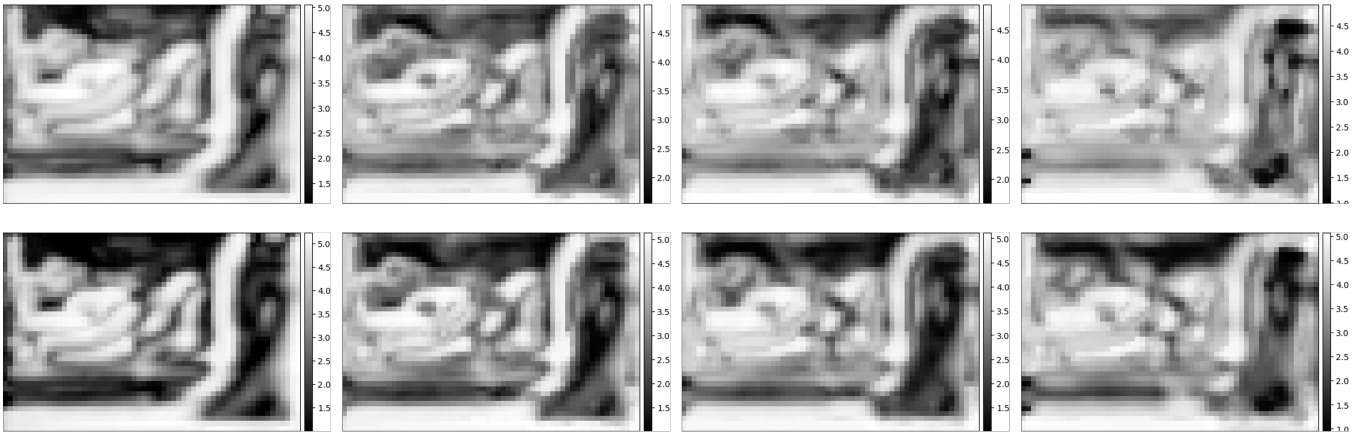
Figure 5: Results obtained by focal stack alignment. The first row displays the result from the 05_castle dataset, the second row displays the result from the 07_toys dataset, and the third row displays the result from the mobo_chip dataset. The images on the left-hand side are generated by superimposing all the frames in the captured focal stack, whereas the images on the right-hand side are produced by overlapping all the frames in the aligned focal stack.

[15] Supasorn Suwajanakorn, Carlos Hernandez, and Steven M. Seitz. "Depth from focus with your mobile phone". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 3497–3506. DOI: 10.1109/CVPR.2015.7298972.

[16] Masahiro Watanabe and Shree K. Nayar. "Telecentric optics for computational vision". In: *Computer Vision — ECCV '96*. Ed. by Bernard Buxton and Roberto Cipolla. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 439–451. ISBN: 978-3-540-49950-3.

[17] Quanbing Zhang and Yanyan Gong. "A Novel Technique of Image-Based Camera Calibration in Depth-from-Defocus". In: *2008 First International Conference on Intelligent Networks and Intelligent Systems*. 2008, pp. 483–486. DOI: 10.1109/ICINIS.2008.95.

[18] Changyin Zhou, Daniel Miau, and Shree K. Nayar. "Focal Sweep Camera for Space-Time Refocusing". In: 2012. URL: https://api.semanticscholar.org/CorpusID:15207045.



$\delta_r = 0.01$ $\delta_r = 0.05$ $\delta_r = 0.1$ $\delta_r = 0.2$

Figure 7: Caption