The "Vertigo Effect" on Your Smartphone: Dolly Zoom via Single Shot View Synthesis

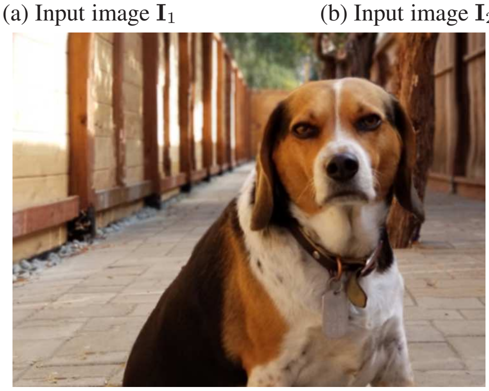
Yangwen Liang Rohit Ranade Shuangquan Wang Dongwoon Bai
Jungwon Lee
Samsung Semiconductor Inc
{liang.yw, rohit.r7, shuangquan.w, dongwoon.bai, jungwon2.lee}@samsung.com

Abstract

Dolly zoom is a technique where the camera is moved either forwards or backwards from the subject under focus while simultaneously adjusting the field of view in order to maintain the size of the subject in the frame. This results in perspective effect so that the subject in focus appears stationary while the background field of view changes. The effect is frequently used in films and requires skill, practice and equipment. This paper presents a novel technique to model the effect given a single shot capture from a single camera. The proposed synthesis pipeline based on camera geometry simulates the effect by producing a sequence of synthesized views. The technique is also extended to allow simultaneous captures from multiple cameras as inputs and can be easily extended to video sequence captures. Our pipeline consists of efficient image warping along with depth–aware image inpainting making it suitable for smartphone applications. The proposed method opens up new avenues for view synthesis applications in modern smartphones.

(a) Input image $\mathbf{I}_1$

(b) Input image $\mathbf{I}_2$



(c) Dolly zoom synthesized image with SDoF by our method

Figure 1: Single camera single shot dolly zoom View Synthesis example. Here (a) is generated from (b) through digital zoom

1. Introduction

The "dolly zoom" effect was first conceived in Alfred Hitchcock's 1958 film "Vertigo" and since then, has been frequently used by film makers in numerous other films. The photographic effect is achieved by zooming in or out in order to adjust the field of view (FoV) while simultaneously moving the camera away or towards the subject. This leads to a continuous perspective effect with the most directly noticeable feature being that the background appears to change size relative to the subject [13]. Execution of the effect requires skill and equipment, due to the necessity of simultaneous zooming and camera movement. It is especially difficult to execute on mobile phone cameras, because of the requirement of fine control of zoom, object tracking and movement.

Previous attempts at automatically simulating this effect required the use of a specialized light field camera [19] while the process in [20, 1] involved capturing images while moving the camera, tracking interest points and then applying a calculated scaling factor to the images. Generation of views from different camera positions given a sequence of images through view interpolation is mentioned in [8] while [10] mentions methods to generate images given a 3D scene. Some the earliest methods for synthesizing images through view interpolation include [6, 23, 31]. More recent methods have applied deep convolutional networks for producing novel views from a single image [30, 17], from multiple input images [28, 9] or for producing new video frames in existing videos [15].

In this paper, we model the effect using camera geometry and propose a novel synthesis pipeline to simulate the effect given a single shot of single or multi–camera captures, where single shot is defined as an image and depth capture collected from each camera at a particular time instant and location. The depth map can be obtained from passive sensing methods, cf. e.g. [2, 11], active sensing methods, cf. e.g. [22, 24] and may also be inferred from a single image through convolutional neural networks, cf. e.g. [5]. The synthesis pipeline handles occlusion areas through depth aware image inpainting. Traditional methods for image inpainting include [3, 4, 7, 26] while others like [16] adopt the method in [7] for depth based inpainting. Recent methods like [21, 14] involve applying convolutional networks for this task. However, since these methods have high complexity, we implement a simpler algorithm suitable for smartphone applications. Our pipeline also includes the application of the shallow depth of field (SDoF) [27] effect for image enhancement. An example result of our method with the camera simulated to move towards the object under focus while changing focal length simultaneously is shown in Figure 1. In this example, Figures 1a, 1b are the input images and Figure 1c is the dolly zoom synthesized image with the SDoF effect applied. Notice that the dog remains in focus and the same size while the background shrinks with an increase in FoV.

The rest of the paper is organized as follows: System model and view synthesis with camera geometry are described in Section 2 while experiment results are given in Section 3. Conclusions are drawn in Section 4.

Notation Scheme In this paper, matrix is denoted as $\mathbf{H}$, and $(\cdot)^{\mathrm{T}}$ denotes transpose. The projection of point $\boldsymbol{P}$, defined as $\boldsymbol{P} = (X, Y, Z)^{\mathrm{T}}$ in $\mathbb{R}^3$, is denoted as point $\boldsymbol{u}$, defined as $\boldsymbol{u} = (x, y)^{\mathrm{T}}$ in $\mathbb{R}^2$. Scalars are denoted as X or x. Correspondingly, $\mathbf{I}$ is used to represent an image. $I(x, y)$, or alternately $I(\boldsymbol{u})$, is the intensity of the image at location (x, y). Similarly, for a matrix $\mathbf{H}$, $H(x, y)$ denotes the element at pixel (x, y) in that matrix. $\mathbf{J}_n$ and $\mathbf{0}_n$ denote the $n \times n$ identity matrix and $n \times 1$ zero vectors. $\mathrm{diag}\{x_1, x_2, \cdots, x_n\}$ denotes a diagonal matrix with elements $x_1, x_2, \cdots, x_n$ on the main diagonal.

2. View Synthesis based on Camera Geometry

Consider two pin-hole cameras A and B with camera centers at locations $\boldsymbol{C}_A$ and $\boldsymbol{C}_B$, respectively. From [12], based on the coordinate system of camera A, the projections of any point $\boldsymbol{P} \in \mathbb{R}^3$ onto the camera image planes are $\left(\boldsymbol{u}_A^{\mathrm{T}}, 1\right)^{\mathrm{T}} = \frac{1}{D_A}\mathbf{K}_A \left[\mathbf{J}_3 \quad \mathbf{0}_3\right] \left(\boldsymbol{P}^{\mathrm{T}}, 1\right)^{\mathrm{T}}$ and $\left(\boldsymbol{u}_B^{\mathrm{T}}, 1\right)^{\mathrm{T}} = \frac{1}{D_B}\mathbf{K}_B \left[\mathbf{R} \quad \boldsymbol{T}\right] \left(\boldsymbol{P}^{\mathrm{T}}, 1\right)^{\mathrm{T}}$ for cameras A and B, respectively. Here, the 2×1 vector $\boldsymbol{u}_X$, the 3×3 matrix $\mathbf{K}_X$, and the scalar D_X are the pixel coordinates on the image plane, the intrinsic parameters, and the depths of $\boldsymbol{P}$ for camera X, $X \in \{A, B\}$, respectively. The 3×3 matrix $\mathbf{R}$ and the 3×1 vector $\boldsymbol{T}$ are the relative rotation and translation of camera B with respect to camera A. In general, the relationship between $\boldsymbol{u}_B$ and $\boldsymbol{u}_A$ can be obtained in closed–form as

$$\begin{pmatrix} \boldsymbol{u}_B \\ 1 \end{pmatrix} = \frac{D_A}{D_B}\mathbf{K}_B\mathbf{R}(\mathbf{K}_A)^{-1}\begin{pmatrix} \boldsymbol{u}_A \\ 1 \end{pmatrix} + \frac{\mathbf{K}_B\boldsymbol{T}}{D_B} \tag{1}$$

where $\boldsymbol{T}$ can also be written as $\boldsymbol{T} = \mathbf{R}\left(\boldsymbol{C}_A - \boldsymbol{C}_B\right)$.

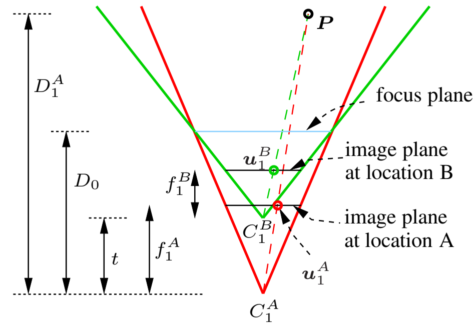


Figure 2: Single camera system setup under dolly zoom

2.1. Single Camera System

Consider the system setup for a single camera under dolly zoom as shown in Figure 2. Here, camera 1 is at an initial position $\boldsymbol{C}_1^A$ with a FoV of θ_1^A and focal length f_1^A (the relationship between the camera FoV θ and its focal length f (in pixel units) may be given as $f = (W/2)/\tan(\theta/2)$ where W is the image width). In order to achieve the dolly zoom effect, we assume that it undergoes translation by a distance t to position $\boldsymbol{C}_1^B$ along with a change in its focal length to f_1^B and correspondingly, a change of FoV to θ_1^B ($\theta_1^B \geq \theta_1^A$). D_1^A is the depth of a 3D point $\boldsymbol{P}$ and D_0 is the depth to the focus plane from the camera 1 at the initial position $\boldsymbol{C}_1^A$. Our goal is to create a synthetic view at location $\boldsymbol{C}_1^B$ from a capture at location $\boldsymbol{C}_1^A$ such that any object in the focus plane is projected at the same pixel location in 2D image plane regardless of camera location. For the same 3D point $\boldsymbol{P}$, $\boldsymbol{u}_1^A$ is its projection onto the image plane of camera 1 at its initial position $\boldsymbol{C}_1^A$ while $\boldsymbol{u}_1^B$ is its projection onto the image plane of camera 1 after it has moved to position $\boldsymbol{C}_1^B$. We make the following assumptions:

1. The translation of the camera center is along the principal axis Z. Accordingly, $\boldsymbol{C_1^A} - \boldsymbol{C_1^B} = \left(0, 0, -t\right)^{\mathrm{T}}$. while the depth of $\boldsymbol{P}$ to the camera 1 at position $\boldsymbol{C}_1^B$ is $D_1^B = D_1^A - t$.

2. There is no relative rotation during camera translation. Therefore, $\mathbf{R}$ is an identity matrix $\mathbf{J}_3$.

3. Assuming there is no shear factor, the camera intrinsic matrix $\mathbf{K}_1^A$ of the camera at location $\boldsymbol{C}_1^A$ can be modeled as [12]

$$\mathbf{K}_1^A = \begin{bmatrix} f_1^A & 0 & u_0 \\ 0 & f_1^A & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad (2)$$

   where $\boldsymbol{u}_0 = (u_0, v_0)^{\mathrm{T}}$ is the principal point in terms of pixel dimensions. Assuming the resulting image resolution did not change, the intrinsic matrix $\mathbf{K}_1^B$ at position $\boldsymbol{C}_1^B$ is related to that at $\boldsymbol{C}_1^A$ through a zooming factor k and can be obtained as $\mathbf{K}_1^B = \mathbf{K}_1^A \mathrm{diag}\{k, k, 1\}$, where $k = f_1^B / f_1^A = (D_0 - t)/D_0$.

4. The depth of $\boldsymbol{P}$ is $D_1^A > t$. Otherwise, $\boldsymbol{P}$ will be excluded on the image plane of camera at position $\boldsymbol{C}_1^B$.

From Eq. 1, we can obtain the closed–form solution for $\boldsymbol{u}_1^B$ in terms of $\boldsymbol{u}_1^A$ as

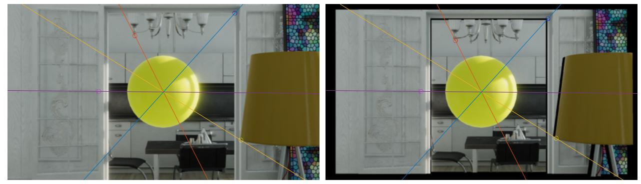
$$\boldsymbol{u}_1^B = \frac{D_1^A (D_0 - t)}{D_0 (D_1^A - t)} \boldsymbol{u}_1^A + \frac{t(D_1^A - D_0)}{D_0 (D_1^A - t)} \boldsymbol{u}_0 \,. \qquad (3)$$

A generalized equation for camera movements along the horizontal and vertical directions along with the translation along the principal axis and change of FoV/focal length is derived in the supplementary.

Let $\mathbf{I}_1$ be the input image from camera 1 and $\mathbf{D}_1$ be the corresponding depth map, so that each pixel $\boldsymbol{u} = (x, y)$, the corresponding depth $D_1(\boldsymbol{u})$ may be obtained. $\mathbf{I}_1$ can now be warped using Eq. 3 using $\mathbf{D}_1$ for a camera translation t to obtain the synthesized image $\mathbf{I}_1^{DZ}$. Similarly, we can warp $\mathbf{D}_1$ with the known t and obtain the corresponding depth $\mathbf{D}_1^{DZ}$. This step is implemented through z-buffering [25] and forward warping. An example is shown in Figure 3.

Epipolar Geometry Analysis In epipolar geometry, pixel movement is along the epipolar lines, which is related by the fundamental matrix between the two camera views. The fundamental matrix $\boldsymbol{F}_1$ relates corresponding pixels in the images for the two views without knowledge of pixel depth information [12]. This is a necessary condition for corresponding points and can be given as $\left(\boldsymbol{x}_1^B\right)^{\mathrm{T}} \mathbf{F}_1 \boldsymbol{x}_1^A = 0$, where $\boldsymbol{x}_1^A = \left((\boldsymbol{u}_1^A)^{\mathrm{T}}, 1\right)^{\mathrm{T}}$ and $\boldsymbol{x}_1^B = \left((\boldsymbol{u}_1^B)^{\mathrm{T}}, 1\right)^{\mathrm{T}}$. From [12], it is easy to show that the fundamental matrix can be obtained as $\mathbf{F}_1 = \begin{bmatrix} 0 & -1 & v_0; 1 & 0 & -u_0; -v_0 & u_0 & 0 \end{bmatrix}$ and the corresponding epipoles and epipolar lines can be obtained accordingly [12], as shown in Figure 3. The epipoles are $\boldsymbol{e}_1^A = \boldsymbol{e}_1^B = (u_0, v_0, 1)^{\mathrm{T}}$ for both locations $\boldsymbol{C}_1^A$ and $\boldsymbol{C}_1^B$ as camera moving along the principal axis [12].

Digital Zoom It is worthy to note that θ_1^A may be a partial FoV of the actual camera FoV θ_1 at initial position $\boldsymbol{C}_1^A$. Straightforward digital zoom can be employed to get the partial FoV image. Assuming the actual intrinsic matrix $\mathbf{K}_1$, the intrinsic matrix $\mathbf{K}_1^A$ for partial FoV can be obtained as $\mathbf{K}_1^A = \mathbf{K}_1 \mathrm{diag}\{k_0, k_0, 1\}$. where $k_0 = f_1^A / f_1 = \tan(\theta_1/2)/\tan(\theta_1^A/2)$. Subsequently, a closed–form equation may be obtained for the zoom pixel coordinates $\boldsymbol{u}_1^A$ in terms of $\boldsymbol{u}_1$ of actual image pixel location (with the camera rotation $\mathbf{R}$ as an identity matrix $\mathbf{J}_3$):



(a) Input image $\mathbf{I}_1$ with FoV $\theta_1^A = 45^o$ (b) Synthesized image $\mathbf{I}_1^{DZ}$ with FoV $\theta_1^B = 50^o$

Figure 3: Single camera image synthesis under dolly zoom. Epipoles (green dots), sample point correspondences (red, blue, yellow, magenta dots) along with their epipolar lines are shown.

$$\boldsymbol{u}_1^A = (f_1^A / f_1) \boldsymbol{u}_1 + \left(1 - (f_1^A / f_1)\right) \boldsymbol{u}_0 \qquad (4)$$

Eq. 4 may be used to digitally zoom $\mathbf{I}_1$ and $\mathbf{D}_1$ to the required FoV θ_1^A.

2.2. Introducing a Second Camera to the System

Applying the synthesis formula from Eq. 3 for a single camera results in many missing and occluded areas as the FoV increases. Some of these areas can be filled using projections from other available cameras with different FoVs. We now introduce a second camera to the system for this purpose. Consider the system shown in Figure 4 where a second camera with focal length f_2 is placed at position $\boldsymbol{C}_2$. As an example, we assume that both cameras are well calibrated [29], i.e. these two cameras are on the same plane and their principal axes are perpendicular to that plane. Let b be the baseline between the two cameras. The projection of point $\boldsymbol{P}$ on the image plane of camera 2 is at pixel location $\boldsymbol{u}_2$.

We once again assume that there is no relative rotation between the two cameras (or that it has been corrected during camera calibration [29]). The translation of the second camera from position $\boldsymbol{C}_2$ to position $\boldsymbol{C}_1^B$ can be given as $\boldsymbol{C}_2 - \boldsymbol{C}_1^B = \left(b, 0, -t\right)^{\mathrm{T}}$. Here, we assume the baseline is on the X-axis, but it is simple to extend to any directions. For the same point $\boldsymbol{P}$, the corresponding depth relationship can be given as $D_1^B = D_2 - t$, where D_2 denotes the depth of $\boldsymbol{P}$ seen by camera 2 at position C_2. Assuming image resolutions are the same, the intrinsic matrix $\mathbf{K}_2$ of camera 2 can be related to the intrinsic matrix of camera 1 at position $\boldsymbol{C}_1^A$ as $\mathbf{K}_2 = \mathbf{K}_1^A \mathrm{diag}\{k', k', 1\}$, where the zooming factor k' can be given as $k' = f_2 / f_1^A = \tan(\theta_1^A/2)/\tan(\theta_2/2)$.

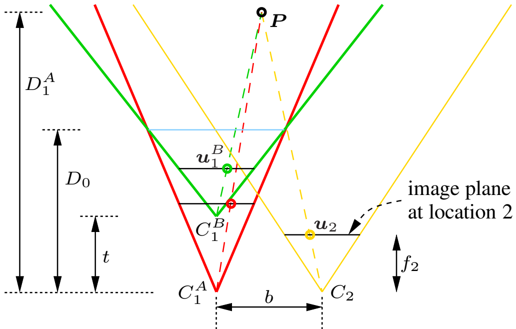


Figure 4: Two cameras system setup under dolly zoom.

(a) Input image $\mathbf{I}_2$ with FoV $\theta_2 = 77^o$ (b) Synthesized image $\mathbf{I}_2^{DZ}$ with FoV $\theta_1^B = 50^o$

Figure 5: Image synthesis for the second camera under dolly zoom. Epipoles (green dots), sample point correspondences (red, blue, yellow, magenta dots) along with their epipolar lines are shown.

A closed–form solution for $\boldsymbol{u}_1^B$ can be obtained as:

$$\boldsymbol{u}_1^B = \frac{D_2 k}{(D_2 - t)k'}(\boldsymbol{u}_2 - \boldsymbol{u}_0) + \boldsymbol{u}_0 + \begin{pmatrix} \frac{bf_1^A k}{D_2 - t} \\ 0 \end{pmatrix}. \quad (5)$$

Let $\mathbf{I}_2$ be the input image from camera 2 and $\mathbf{D}_2$ be the corresponding depth map. $\mathbf{I}_2$ can now be warped using Eq. 5 with $\mathbf{D}_2$ for a camera translation t to obtain the synthesized image $\mathbf{I}_2^{DZ}$. We once again use forward warping with z–buffering for this step. An example is shown in Figure 5. This derivation can be easily extended to include any number of additional cameras to the system.

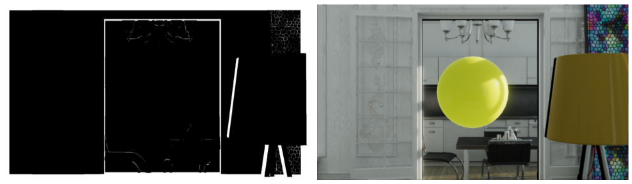
A generalized equation for camera movements along the horizontal and vertical directions along with the translation along the principal axis and change of FoV/focal length is derived in the supplementary for this case as well.

Epipolar Geometry Analysis Similar to single camera case, we can derive the fundamental matrix $\mathbf{F}_2$ in close–form

$$\mathbf{F}_2 = \begin{bmatrix} 0 & -t & tv_0 \\ t & 0 & bf_1^A k' - tu_0 \\ -tv_0 & tu_0 - bf_1^A k & bf_1^A v_0 (k - k') \end{bmatrix} \quad (6)$$

such that pixel location relationship $\left(\boldsymbol{x}_1^B\right)^{\mathrm{T}} \mathbf{F}_2 \boldsymbol{x}_2 = 0$ is satisfied. Here, $\boldsymbol{x}_2 = \left(\boldsymbol{u}_2^{\mathrm{T}}, 1\right)^{\mathrm{T}}$ is the homogeneous representation of pixels of camera at location 2. Therefore, the corresponding epipoles are $\boldsymbol{e}_1^B = \left(u_0 - bf_1^A k/t, v_0, 1\right)^{\mathrm{T}}$ and $\boldsymbol{e}_2 = \left(u_0 - bf_1^A k'/t, v_0, 1\right)^{\mathrm{T}}$ for cameras at locations $\boldsymbol{C}_1^B$ and $\boldsymbol{C}_2$, respectively. Also, epipolar lines can be obtained accordingly [12] as shown in Figure 5.

(a) Synthesized image $\mathbf{I}_1^{DZ}$ with FoV $\theta_1^B = 50^o$ (b) Synthesized image $\mathbf{I}_2^{DZ}$ with FoV $\theta_2^B = 50^o$



(c) Binary mask $\mathbf{B}$ (d) Synthesized fused image $\mathbf{I}_F$

Figure 6: Image fusion

2.3. Image Fusion

We now intend to use the synthesized image $\mathbf{I}_2^{DZ}$ from the second camera to fill in missing/occlusion areas in the synthesized image $\mathbf{I}_1^{DZ}$ from the first camera. This is achieved through image fusion with the following steps:

1. The first step is to identify missing areas in the synthesized view $\mathbf{I}_1^{DZ}$. Here, we implement a simple scheme given below to create a binary mask $\mathbf{B}$ by checking the validity of $\mathbf{I}_1^{DZ}$ at each pixel location (x, y):

$$B(x,y) = \begin{cases} 1, & I_1^{DZ}(x,y) \in \boldsymbol{O}_1^{m,c} \\ 0, & I_1^{DZ}(x,y) \notin \boldsymbol{O}_1^{m,c} \end{cases} \quad (7)$$

where $\boldsymbol{O}_1^{m,c}$ denotes a set of missing/occluded pixels for $\mathbf{I}_1^{DZ}$ due to warping.

2. With the binary mask $\mathbf{B}$, the synthesized images $\mathbf{I}_1^{DZ}$ and $\mathbf{I}_2^{DZ}$ are fused to generate $\mathbf{I}_F$:

$$\mathbf{I}_F = \mathbf{B} \cdot \mathbf{I}_2^{DZ} + (\mathbf{1} - \mathbf{B}) \cdot \mathbf{I}_1^{DZ} \quad (8)$$

where $\cdot$ is element-wise matrix product. The depths for the synthesized view $\mathbf{D}_1^{DZ}$ and $\mathbf{D}_2^{DZ}$ are also fused in a similar manner to obtain $\mathbf{D}_F$. An example is shown in Figure 6. In this example, the FoV of the second camera is greater than that of the first camera (i.e. $\theta_2 > \theta_1^A$) in order to fill larger missing area. For image fusion, we can also apply more advanced methods which are capable of handling photometric differences between input images, e.g. Poisson fusion [18].

2.4. Depth Aware Image Occlusion Handling

In order to handle occlusions, for each synthesized dolly zoom view, we identify occlusion areas and fill them in using neighboring information for satisfactory subjective viewing. Occlusions occur due to the nature of the camera movement and depth discontinuity along the epipolar line [12]. Therefore, one constraint in filling occlusion areas is that whenever possible, they should be filled only with the background and not the foreground.

Occlusion Area Identification The first step is to identify occlusion areas. Let $\mathbf{I}_F$ be the generated view after image fusion. Let $\mathbf{M}$ be a binary mask depicting occlusion areas. Similar to section 2.3, $\mathbf{M}$ is simply generated by checking the validity of $\mathbf{I}_F$ at each pixel location (x, y).

$$M(x,y) = \begin{cases} 1, & I_F(x,y) \in \boldsymbol{O}_F^c \\ 0, & I_F(x,y) \notin \boldsymbol{O}_F^c \end{cases} \tag{9}$$

where $\boldsymbol{O}_F^c$ denotes a set of occluded pixels for $\mathbf{I}_F$ after image fusion in Section 2.3.

Depth Hole–Filling A critical piece of information is the fused depth $\mathbf{D}_F$ for the synthesized view which allows us to distinguish between foreground and background. $\mathbf{D}_F$ will also have holes due to occlusion. If we intend to use the depth for image hole–filling, we need to first fill the holes in the depth itself. We implement a simple nearest neighbor hole filling scheme described in Algorithm 1.

Algorithm 1: Depth map hole filling

Input : Fused depth $\mathbf{D}_F$, dimensions (width W and height H)
Output: The hole filled depth $\overline{\mathbf{D}_F}$
1. Initialize $\overline{\mathbf{D}_F} = \mathbf{D}_F$.
for $x = 1$ **to** H **do**
  for $y = 1$ **to** W **do**
    if $M(x, y) = 1$ **then**
      2.1) Find four nearest neighbors (left, right, bottom, top).
      2.2) Find the neighbor with the maximum value (d_{max}), since we intend to fill in the missing values with background values.
      2.3) Set $\overline{D_F}(x, y) = d_{max}$
    end
  end
end

Depth–Aware Image Inpainting Hole filling for the synthesized view needs to propagate from the background towards the foreground. The hole filling strategy is described in Algorithm 2.

Algorithm 2: Image hole filling

Input : Synthesized view $\mathbf{I}_F$, Synthesized depth $\overline{\mathbf{D}_F}$, Occlusion mask $\mathbf{M}$, depth segment mask $\mathbf{M}_{prev}$ initialized to zeros and dimensions (width W and height H)
Output: The hole filled synthesized view $\overline{\mathbf{I}_F}$
1. Initialize: $\overline{\mathbf{I}_F} = \mathbf{I}_F$
2. Determine all unique values in $\overline{\mathbf{D}_F}$. Let $\mathbf{d}^u$ be the array of unique values in the ascending order and S be the number of unique values.
for $s = S$ **to** 2 **do**
  3.1) Depth mask $\mathbf{D}_s$ corresponding to the depth step:

$$\mathbf{D}_s = (\overline{\mathbf{D}}_F > d^u(s-1)) \& (\overline{\mathbf{D}}_F \leq d^u(s))$$

  where $>$, $\leq$ and & are the element-wise matrix greater than, less than or equal to and AND operations.
  3.2) Image segment $\mathbf{I}_s$ corresponding to the depth mask:

$$\mathbf{I}_s = \mathbf{I}_F \cdot \mathbf{D}_s$$

  where $\cdot$ is element-wise matrix product.
  3.3) Current Occlusion mask for the depth step:

$$\mathbf{M}_{curr} = \mathbf{M} \cdot \mathbf{D}_s$$

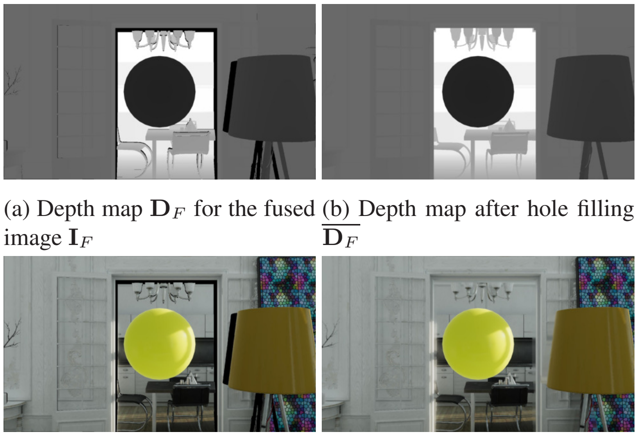
  3.4) Update $\mathbf{M}_{curr}$ with previous mask

$$\mathbf{M}_{curr} = \mathbf{M}_{curr} \parallel \mathbf{M}_{prev}$$

  where $\parallel$ is element-wise matrix OR condition.
  for $x = 1$ **to** H **do**
    for $y = 1$ **to** W **do**
      if $M_{curr}(x, y) = 1$ **then**
        3.5.1) Find nearest valid pixels on the same row $I_s(x', y')$, where (x', y') is the location of the valid pixel.
        3.5.2) Update value of $\overline{I}_F(x, y) = I_s(x', y')$
        3.5.3) Update $M_{curr}(x, y) = 0$
        3.5.4) Update $M(x, y) = 0$
      end
    end
  end
  3.6) Propagate the current occlusion mask to the next step:

$$\mathbf{M}_{prev} = \mathbf{M}_{curr}$$

end
4. Apply simple low pass filtering on the filled in occluded areas in $\overline{\mathbf{I}_F}$.



(a) Depth map $\mathbf{D}_F$ for the fused image $\mathbf{I}_F$ (b) Depth map after hole filling $\overline{\mathbf{D}_F}$

(c) Synthesized fused image $\mathbf{I}_F$ (d) Synthesized fused image after hole filling $\overline{\mathbf{I}_F}$

Figure 7: Occlusion handling

This strategy is implemented in a back–to–front order with the intuition being that holes in the image should be filled in from parts of the image at the same depth or the next closest depth. In this simple hole filling algorithm, we search for valid image pixels along the same row but this could also be extended to finding more than one valid pixels in both the horizontal and vertical directions or using epipolar analysis described in Sections 2.1 and 2.2 to define search directions. The results of the hole filling process are shown in Figure 7.

2.5. Shallow Depth of Field (SDoF)

After view synthesis and occlusion handling, we can apply the shallow depth of field (SDoF) effect to $\overline{\mathbf{I}_F}$. This effect involves the application of depth–aware blurring. The diameter c of the blur kernel on the image plane is called the circle of confusion (CoC). Assuming a thin lens camera model [12], the relation betwen c, lens aperture A, magnification factor m, distance to an object under focus D_0 and another object at distance D can be given as [25] $c = Am(|D - D_0|)/D$. Under the dolly zoom condition, the magnification factor m and the relative distance $|D - D_0|$ remains constant. However, after a camera translation of t, the diameter of the CoC changes to:

$$c(t) = Am\frac{|D - D_0|}{(D - t)} = c(0)\frac{D}{(D - t)}\ , \qquad (10)$$

where $c(t)$ is the CoC for an object at depth D and the camera translation t. The detailed derivation can be found in the supplementary. The usage of SDoF effect is two–fold: 1) enhanced viewer attention to the objects in focus, and 2) hide imperfections due to image warping, image fusion and hole filling steps.

2.6. Dolly Zoom View Synthesis Pipeline

Single Camera Single Shot View Synthesis The single shot single camera dolly zoom synthesis pipeline is shown in Figure 8 and described below:

1. The input is the image $\mathbf{I}$ with FoV θ, its depth map $\mathbf{D}$ and the known intrinsic matrix $\mathbf{K}$.
2. We apply digital zoom to both $\mathbf{I}$ and $\mathbf{D}$ according to Eq. 4 (described in Section 2.1) through inverse warping to a certain angle θ_1 (in the example experiments, θ_1 is set to 30^o) to obtain the zoomed-in image $\mathbf{I}_1$ and the corresponding depth map $\mathbf{D}_1$.
3. The original input image $\mathbf{I}$, depth map $\mathbf{D}$ and intrinsic matrix $\mathbf{K}$ are re-used as $\mathbf{I}_2$, $\mathbf{D}_2$ and $\mathbf{K}_2$ respectively.
4. A synthesized image $\mathbf{I}_1^{DZ}$ and its depth $\mathbf{D}_1^{DZ}$ is produced from $\mathbf{I}_1$ and $\mathbf{D}_1$ with Eq. 3 through forward warping and z-buffering for a given t.
5. A synthesized image $\mathbf{I}_2^{DZ}$ and its depth $\mathbf{D}_2^{DZ}$ is produced from $\mathbf{I}_2$ and $\mathbf{D}_2$ with Eq. 5 through forward warping and z-buffering for the given t. The baseline b is set to 0 for this case.
6. The synthesized images $\mathbf{I}_1^{DZ}$ and $\mathbf{I}_2^{DZ}$ are fused together (as described in Section 2.3) to form the fused image $\mathbf{I}_F$ while the synthesized depth maps $\mathbf{D}_1^{DZ}$ and $\mathbf{D}_2^{DZ}$ are similarly fused together to form the fused depth map $\mathbf{D}_F$.
7. Occlusion areas in $\mathbf{I}_F$ (and $\mathbf{D}_F$) are handled (according to Section 2.4) to obtain $\overline{\mathbf{I}_F}$ (and $\overline{\mathbf{D}_F}$).
8. The shallow depth of field effect is applied to $\overline{\mathbf{I}_F}$ to obtain the final dolly zoom synthesized image $\overline{\mathbf{I}_F^{DZ}}$.

A restriction of this setup is that the maximum FoV for the synthesized view is limited to θ_1.

Extending the pipeline for multiple camera inputs The single camera single shot dolly zoom synthesis pipeline may be extended to input images captured from multiple cameras at the same time instant with minor modifications. Consider a dual camera setup, where the inputs are the image $\mathbf{I}_1$ with FoV θ_1, its depth map $\mathbf{D}_1$ and the known intrinsic matrix $\mathbf{K}_1$ from the first camera and correspondingly,the image $\mathbf{I}_2$ with FoV θ_2, its depth map $\mathbf{D}_2$ and the known intrinsic matrix $\mathbf{K}_2$ from the second camera. In this case, the application of digital zoom in Step 2 of the single camera pipeline is no longer required. Instead, we only apply Steps 4 – 8 with the baseline b set to the representative value, to obtain the synthesized view $\overline{\mathbf{I}_F^{DZ}}$ for the dual camera case. A restriction of such a setup is that the maximum FoV for the synthesized view is now limited to θ_2 (and in general, to the FoV of the camera with the largest FoV in the multi-camera system).

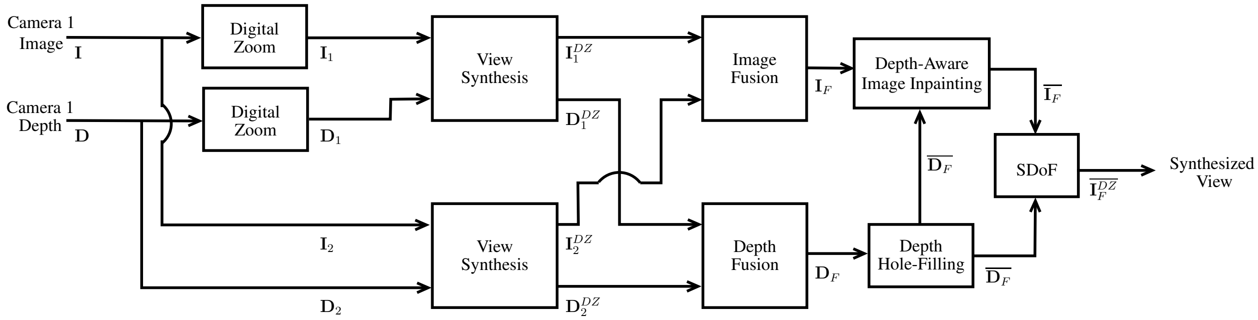


Figure 8: Single camera single shot synthesis pipeline

3. Experiment Results

3.1. Datasets

Synthetic Dataset We generated a synthetic image dataset using the commercially available graphics software Unity 3D. For the experiment, we assume a dual-camera collinear system with the following parameters: Camera 1 with FoV $\theta_1 = 45^o$ and Camera 2 with FoV $\theta_2 = 77^o$ This setup is simulated in the Unity3D software. In this synthetic dataset, each image set includes: $\mathbf{I}_1$ from Camera 1 and $\mathbf{I}_2$ from Camera 2 , the depth maps $\mathbf{D}_1$ for Camera 1, $\mathbf{D}_2$ for Camera 2, and the intrinsic matrices $\mathbf{K}_1$ and $\mathbf{K}_2$ for Camera's 1 and 2 respectively. In addition, each image set also includes the ground truth dolly zoom views which are also generated with Unity3D for objective comparisons.

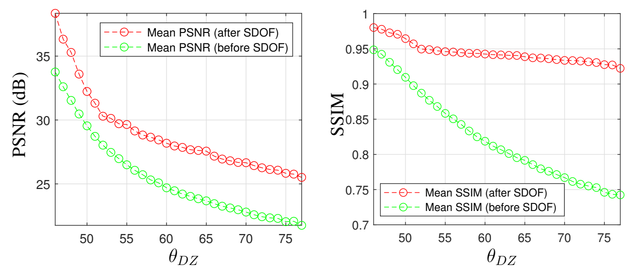
Smartphone Dataset We also created a second dataset with dual camera images from a representative smartphone device. For this dataset, the depth was estimated using a stereo module so that each image set includes $\mathbf{I}$ from Camera 1 with a FoV $\theta = 45^o$, its depth map $\mathbf{D}$ and the intrinsic matrix $\mathbf{K}$.

3.2. Experiment Setup

From the input images, we generate a sequence of images using the synthesis pipeline described in Section 2. For each image set, the depth to the object under focus (D_0) is set manually. The relationship between the dolly zoom camera translation distance t, the required dolly zoom camera FoV θ_{DZ} and the initial FoV θ_1 of $\mathbf{I}_1$ can be obtained from Section 2.1 as:

$$t = D_0 \frac{\tan(\theta_{DZ}/2) - \tan(\theta_1/2)}{\tan(\theta_{DZ}/2)} \,. \tag{11}$$

Initializing the dolly zoom angle θ_{DZ} to θ_1, we increment it by a set amount δ up to a certain maximum angle set to θ_2. For each increment, we obtain the corresponding distance t with Eq. 11. We then apply the synthesis pipeline described in Section 2.6 to obtain the synthesized image $\overline{\mathbf{I}_F^{DZ}}$ for that increment. The synthesized images for all the increments are then compiled to form a single sequence.



(a) Objective metric – PSNR (b) Objective metric – SSIM

Figure 9: Quantitative evaluation

3.3. Quantitative Evaluation

The synthesis pipeline modified for a dual camera input as described in Section 2.6 is applied to each image set in the synthetic dataset. In order to produce a sequence of images, we initialize the dolly zoom angle $\theta_{DZ} = 45^o$ and increment it with a $\delta = 1^o$ up to a maximum angle of $\theta_2 = 77^o$. The dolly zoom camera translation distance t at each increment is calculated according to Eq. 11. We then objectively measure the quality of our view synthesis by computing the peak signal-to-noise ratio (PSNR) and the structural similarity index (SSIM) for each synthesized view against the corresponding ground truth image at each increment, before and after the application of the SDoF effect. The mean metric values for each increment are then computed across all the image sets in the synthetic dataset and are shown in Figure 9. As the dolly zoom angle θ_{DZ} increases, the area of the image that need to be inpainted due to occlusion increases, which corresponds to the drop in PSNR and SSIM.

3.4. Qualitative Evaluation

Figure 10 shows the results for the dual camera input synthesis pipeline applied to the image sets in the synthetic dataset. The input images $\mathbf{I}_1$ and $\mathbf{I}_2$ are used to produce the dolly zoom image $\overline{\mathbf{I}_F}$ while the right most column shows the corresponding ground truth dolly zoom image. Both $\overline{\mathbf{I}_F}$ and

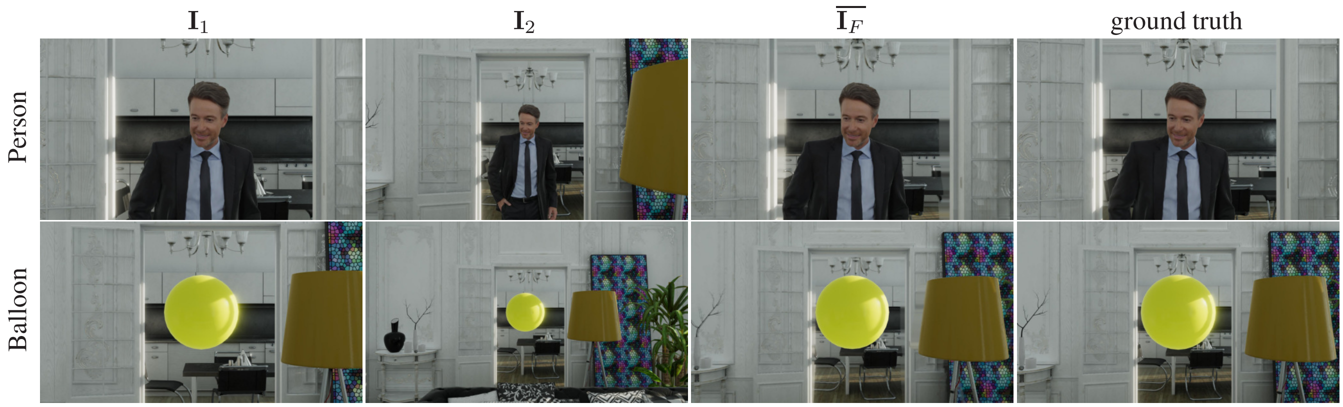


Figure 10: Dual camera dolly zoom view synthesis with synthetic data set.

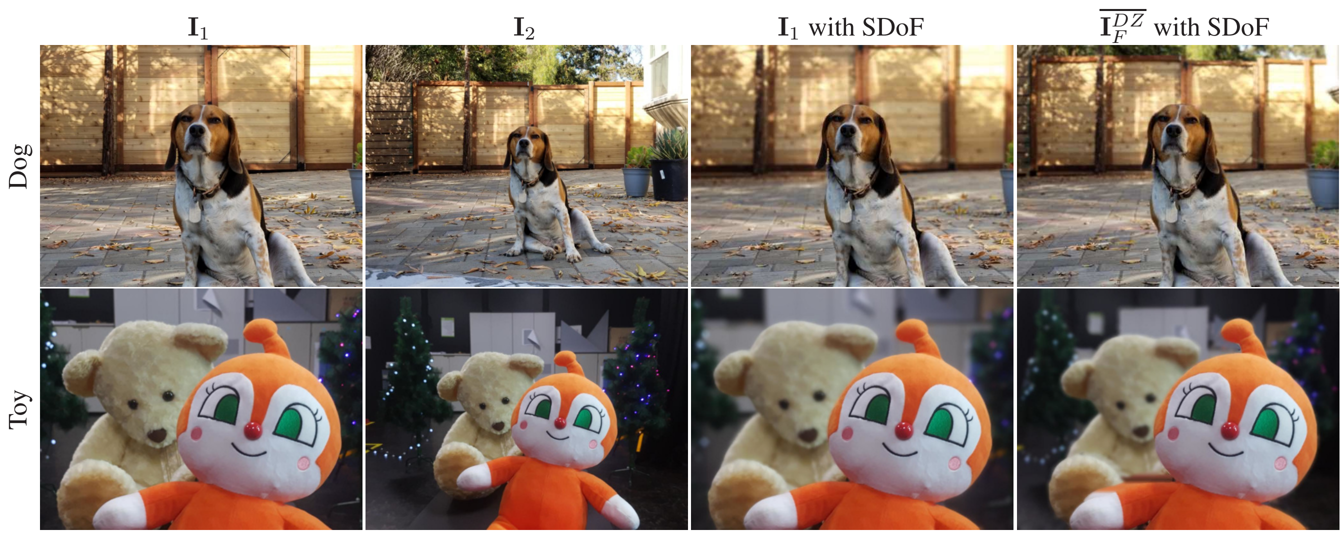


Figure 11: Single camera single shot dolly zoom view synthesis with smartphone data set.

the ground truth are shown here before application of the SDoF effect. Figure 11 shows the results for the single camera single shot synthesis pipeline described in Section 2.6 applied to image sets in the smartphone dataset. Here, the input images $\mathbf{I}_1$ and $\mathbf{I}_2$ (formed from image $\mathbf{I}$ of each image set as described in Section 2.6, Steps 2 – 3) are used to synthesize the dolly zoom image $\overline{\mathbf{I}_F^{DZ}}$ (shown after the application of the SDoF effect). For comparison, we also show $\mathbf{I}_1$ with the SDoF effect applied. Under dolly zoom, objects in the background appear to undergo depth-dependent movement, the objects under focus stay the same size while the background FoV increases. This effect is apparent in our view synthesis results. In both Figures 10 and 11, the foreground objects (balloon, person, dog and toy) remain in focus and of the same size, the background objects are warped according to their depths while the synthesized images ($\overline{\mathbf{I}_F}$ and $\overline{\mathbf{I}_F^{DZ}}$) have a larger background FoV than $\mathbf{I}_1$.

4. Conclusion and Future Work

We have presented a novel modelling pipeline based on camera geometry to synthesize the dolly zoom effect. The synthesis pipeline presented in this paper can be applied to single camera or multi-camera image captures (where the cameras may or may not be on the same plane) and to video sequences. Generalized equations for camera movement not just along the principal axis and change of focal length/FoV but also, along the horizontal or vertical directions have been derived in the supplementary as well. The focus of future work will be on advanced occlusion handling schemes to provided synthesized images with subjectively greater image quality.

References

[1] Abhishek Badki, Orazio Gallo, Jan Kautz, and Pradeep Sen. Computational zoom: a framework for post–capture im-

age composition. *ACM Transactions on Graphics (TOG)*, 36(4):46, 2017.

[2] Jonathan T. Barron, Andrew Adams, YiChang Shih, and Carlos Hernández. Fast bilateral–space stereo for synthetic defocus. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[3] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '00)*, pages 417–424, 2000.

[4] Marcelo Bertalmio, Luminita Vese, Guillermo Sapiro, and Stanley Osher. Simultaneous structure and texture image inpainting. *IEEE Transactions on Image Processing*, 12(8):882–889, 2003.

[5] Po-Yi Chen, Alexander H. Liu, Yen-Cheng Liu, and Yu-Chiang Frank Wang. Towards scene understanding: Unsupervised monocular depth estimation with semantic-aware representation. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[6] Shenchang Eric Chen and Lance Williams. View interpolation for image synthesis. *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '93)*, pages 279–288, 1993.

[7] Antonio Criminisi, Patrick Pérez, and Kentaro Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on Image Processing*, 13(9):1200–1212, 2004.

[8] Maha El Choubassi, Yan Xu, Alexey M. Supikov, and Oscar Nestares. View interpolation for visual storytelling. US Patent US20160381341A1, June, 2015.

[9] John Flynn, Michael Broxton, Paul Debevec, Matthew DuVall, Graham Fyffe, Ryan Overbeck, Noah Snavely, and Richard Tucker. Deepview: View synthesis with learned gradient descent. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[10] Orazio Gallo, Jan Kautz, and Abhishek Haridas Badki. System and methods for computational zoom. US Patent US20160381341A1, December, 2016.

[11] Rahul Garg, Neal Wadhwa, Sameer Ansari, and Jonathan T. Barron. Learning single camera depth estimation using dual-pixels. *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

[12] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Wiley, 2007.

[13] Steven Douglas Katz. *Film directing shot by shot: visualizing from concept to screen*. Gulf Professional Publishing, 1991.

[14] Hongyu Liu, Bin Jiang, Yi Xiao, and Chao Yang. Coherent semantic attention for image inpainting. *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[15] Ziwei Liu, Raymond A. Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala. Video frame synthesis using deep voxel flow. *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[16] Patrick Ndjiki-Nya, Martin Koppel, Dimitar Doshkov, Haricharan Lakshman, Philipp Merkle, Karsten Muller, and Thomas Wiegand. Depth image-based rendering with advanced texture synthesis for 3-d video. *IEEE Transactions on Multimedia*, 13(3):453–465, 2011.

[17] Simon Niklaus, Long Mai, Jimei Yang, and Feng Liu. 3d ken burns effect from a single image. *ACM Transactions on Graphics (TOG)*, 38(6):1–15, 2019.

[18] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. *ACM Transactions on Graphics (TOG)*, 22(3):313–318, July 2003.

[19] Colvin Pitts, Timothy James Knight, Chia-Kai Liang, and Yi-Ren Ng. Generating dolly zoom effect using light field data. US Patent US8971625, March, 2015.

[20] Timo Pekka Pylvanainen and Timo Juhani Ahonen. Method and apparatus for automatically rendering dolly zoom effect. WO Patent Application WO2014131941, September, 2014.

[21] Yurui Ren, Xiaoming Yu, Ruonan Zhang, Thomas H. Li, Shan Liu, and Ge Li. Structureflow: Image inpainting via structure-aware appearance flow. *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[22] Daniel Scharstein and Richard Szeliski. High–accuracy stereo depth maps using structured light. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2003.

[23] Steven M Seitz and Charles R Dyer. View morphing. *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '96)*, pages 21–30, 1996.

[24] Shuochen Su, Felix Heide, Gordon Wetzstein, and Wolfgang Heidrich. Deep end–to–end time–of–flight imaging. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[25] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer, 2011.

[26] Alexandru Telea. An image inpainting technique based on the fast marching method. *Journal of Graphics Tools*, 9(1):23–34, 2004.

[27] Neal Wadhwa, Rahul Garg, David E. Jacobs, Bryan E. Feldman, Nori Kanazawa, Robert Carroll, Yair Movshovitz-Attias, Jonathan T. Barron, Yael Pritch, and Marc Levoy. Synthetic depth–of–field with a single–camera mobile phone. *ACM Transactions on Graphics (TOG)*, 37(4):64:1–64:13, July 2018.

[28] Zexiang Xu, Sai Bi, Kalyan Sunkavalli, Sunil Hadap, Hao Su, and Ravi Ramamoorthi. Deep view synthesis from sparse photometric images. *ACM Transactions on Graphics (TOG)*, 38(4):76, 2019.

[29] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, Nov 2000.

[30] Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. View synthesis by appearance flow. *European Conference on Computer Vision (ECCV)*, pages 286–301, 2016.

[31] C Lawrence Zitnick, Sing Bing Kang, Matthew Uyttendaele, Simon Winder, and Richard Szeliski. High-quality video view interpolation using a layered representation. *ACM Transactions on Graphics (TOG)*, 23(3):600–608, 2004.