# Computational Zoom: A Framework for Post-Capture Image Composition (Supplementary Material)

## 1 PLANE-INDUCED HOMOGRAPHY

Given two cameras $c_i$ and $c_k$, with projection matrices $P_i = K_i[R_i|t_i]$ and $P_k = K_k[R_k|t_k]$, the homography induced by a 3D plane $\pi = [n^T \; d]$ from $c_i$ to $c_k$ is given by [Hartley and Zisserman 2004]:

$$\begin{aligned} H^{\pi}_{i\to k} &= K_k\Big(R - \frac{1}{d}tn^T\Big)K_i^{-1}, \\ R &= R_k R_i^T, \\ t &= t_k - Rt_i, \end{aligned} \tag{1}$$

where $R$ and $t$ are the relative rotation and translation of camera $c_k$ with respect to $c_i$.

## 2 DEFINING MULTI-PERSPECTIVE CAMERAS

In this section, we describe in more detail how multiple pin-hole cameras along with a few 3D planes, which we call dolly planes, could be combined to form a single multi-perspective camera. To achieve this, multiple pin-hole cameras are first transformed using plane-induced homographies before they are merged to define a multi-perspective camera. We will first describe the effect of applying the plane-induced homography to a pin-hole camera and why this operation is important in the context of computational zoom. Later, we will describe how to define a multi-perspective camera using multiple pin-hole cameras.

### 2.1 Effect of applying a plane-induced homography

Let us assume that we have two cameras with projection matrices $P_1 = K_1[R \mid t]$ and $P_2 = K_2[I \mid 0]$ and a dolly plane defined as $\xi_1 = [n^T \; d]$. We first show that applying a homography given by $H^{\xi_1}_{2\to 1}$ to the second camera aligns the images of scene point that lies on $\xi_1$ in both cameras. Assume $\widetilde{X}$ to be a 3D scene point on the plane $\xi_1$ and $X$ its homogeneous co-ordinate representation. Then we have $n^T\widetilde{X} + d = 0$. The images of $\widetilde{X}$ under projections $P_1$ and $P_2$ are given by $x_1 = P_1X = K_1(R\widetilde{X} + t)$ and $x_2 = P_2X = K_2\widetilde{X}$ respectively. Let $x_2' = \widehat{P}_2X$ where $\widehat{P}_2 = H^{\xi_1}_{2\to 1}P_2$. Below we show that $x_2' = x_1$:

$$\begin{aligned} x_2' &= H^{\xi_1}_{2\to 1}P_2X \\ &= H^{\xi_1}_{2\to 1}K_2\widetilde{X} \\ &= K_1\Big(R - \frac{1}{d}tn^T\Big)K_2^{-1}K_2\widetilde{X} \\ &= K_1\Big(R\widetilde{X} - \frac{1}{d}tn^T\widetilde{X}\Big) \\ &= K_1(R\widetilde{X} + t) \\ &= x_1. \end{aligned}$$

Hence, images of an object that lies on the plane $\xi_1$ are aligned when imaged using $P_1$ and $\widehat{P}_2$. Modifying the camera projections this way allows us to define different projection operators in front and beyond the dolly plane while making sure that such transition does not cause any alignment artifacts in the multi-perspective image. Note that this relation holds true regardless of the camera configurations used.

Next we analyze how this operation of modifying the camera projection using plane-induced homography affects the image of scene points that do not lie on the dolly-plane. To explain this in context of our computational zoom application, we specialize the extrinsic parameters of the cameras. We set up two cameras, $P_1 = K_1[R \mid t]$ and $P_2 = K_2[I \mid 0]$, such that $R = I$ and $t = [0\;0\;-\alpha]^T$. Without loss of generality, we assume the intrinsic matrices $K_1$ and $K_2$ to be identity. We define the dolly plane to be aligned with z-axis as $\xi_1 = [0\;0\;1\;-\beta]^T$. Assume a 3D scene point given by $\widetilde{X} = [x_0\; y_0\; z_0]$ that may not lie on the plane $\xi_1$ and $X$ is its homogeneous co-ordinate representation.

We now describe the relationship between the images of scene point $\widetilde{X}$ under projection matrices $P_1$ and $\widehat{P}_2$ as denoted by $x_1$ and $x_2'$ respectively:

$$\begin{aligned} x_1 &= P_1X = \frac{1}{z_0 - \alpha}\,[x_0\; y_0\; 1], \\ x_2' &= H^{\xi_1}_{2\to 1}P_2X = \frac{1}{z_0\Big(1 - \frac{\alpha}{\beta}\Big)}[x_0\; y_0\; 1], \\ x_2' &= \frac{\Big(1 - \frac{\alpha}{z_0}\Big)}{\Big(1 - \frac{\alpha}{\beta}\Big)}x_1 = m\,x_1. \end{aligned}$$

Note that for the camera configuration discussed here, the epipolar lines are always radial and the epipole lies at the intersection of the principal axis and the image plane. Also, the corresponding epipolar lines in two images align. As seen from the above equations, $x_2'$ is multiplied by a factor $m$ with respect to point $x_1$. The factor $m$ indicates how far the point $x_2'$ lies from the principal point, as compared to point $x_1$. Note that the principal point here is simply $[0\;0\;1]^T$, as represented in 2D projective space.

When $z_0 = \beta$, that is the scene point $\widetilde{X}$ lies on the dolly plane, then $m = 1$. This implies that the imaged points align. When $\widetilde{X}$ lies beyond the dolly plane, $m > 1$, which implies that $x_2'$ is farther from the principal point as compared to $x_1$. This means that the objects that lie beyond the dolly plane are magnified when imaged using projection matrix $\widehat{P}_2$ as compared to $P_1$. On the other hand, for the objects that lie in front of the dolly plane, $m < 1$ and hence they are shrunk when imaged under projection matrix $\widehat{P}_2$ as compared to $P_1$. Note that the amount of magnification is the function of $\alpha$, $\beta$ as well as the depth of scene point $z_0$. The magnification factor deviates away from $m = 1$ as distance of objects from the dolly plane increases.

To summarize, for the configuration discussed here, images of objects that lie on the dolly plane are aligned when projected under projections $\widehat{P}_2$ and $P_1$. Objects that lie beyond the dolly plane are magnified under projection $\widehat{P}_2$ as compared to $P_1$ while the objects

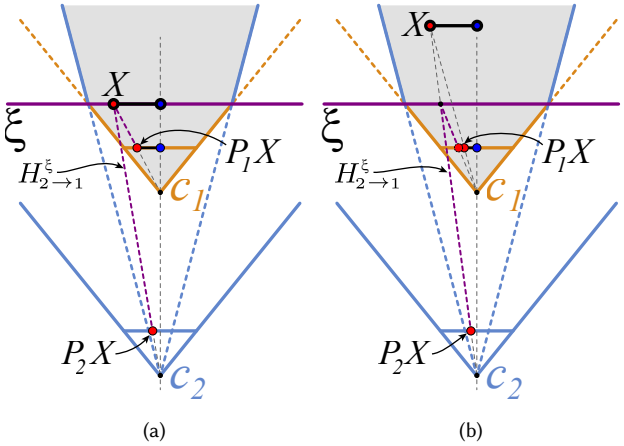


Fig. 1. We can achieve a *short-long* multi-persepective configuration (shown by the shaded region) by combining information taken by cameras at $c_1$ and $c_2$ using the homography between them induced by plane $\xi_1$ (written as $H_{2\to1}^{\xi_1}$). (a) For points $X$ on the plane, their image in camera $c_2$ (given by $P_2X$) will be mapped by this homography (shown with the dashed purple line) to the same point in camera $c_1$, i.e., $P_1X = H_{2\to1}^{\xi_1}P_2X$. Thus, objects on the dolly plane will remain the same size in both images. (b) On the other hand, objects behind the plane imaged by the camera at $c_2$ will be mapped by the homography $H_{2\to1}^{\xi_1}$ to be *larger* when projected back to $c_1$. Therefore, objects behind the dolly plane taken from camera at $c_2$ will look larger in the final result. A similar explanation can be made for a *long-short* configuration. Hence, projection $\widehat{P}_2 = H_{2\to1}^{\xi_1}P_2$ can be used to project points imaged by camera $c_2$ into the first camera in order to merge them together.

that lie in front of dolly plane are shrunk under projection $\widehat{P}_2$ as compared to $P_1$.

We could now define our multi-perspective camera using the two pin-hole cameras given by projection matrices $P_1$ and $\widehat{P}_2$. We can independently set different projections in front and beyond the dolly plane to control the magnification of objects in front and beyond the dolly plane. The images of objects that lie on the dolly plane remain unchanged under projection $P_1$ as compared to projection $\widehat{P}_2$.

In the above analysis, we demonstrated the need for using plane-induced homographies to modify the pin-hole projections so that we could combine them to form a multi-perspective camera. In Fig. 1 we show intuitively through simple ray diagrams how the plane-induced homography helps in defining a multi-perspective camera projection.

For the analysis done in this section, we assumed a very specific camera configuration where there is no relative rotation between the two cameras and the relative translation is only along z-axis. In practice, however, for our computational zoom application we form a multi-perspective camera definition using the images captured by a hand-held camera that moves into the scene. Hence, there is small relative camera rotation and the relative camera translations is only approximately along the z axis. The above analysis holds approximately even in this case. The plane-induced homography still aligns the images of objects that lie on the dolly plane. This ensures that discontinuities do not occur when we use different projection matrices to project scene points in front and beyond the dolly plane. For the objects that lie away from the dolly plane we still get approximately the same behavior where objects beyond the dolly plane appear to be magnified when projected using $\widehat{P}_2$ as compared to $P_1$ and a reverse effect for the objects that lie in front of the dolly plane.

## 2.2 Defining multi-perspective cameras using plane-induced homographies

Using the two projection matrices $P_1$ and $\widehat{P}_2$, along with dolly-plane $\xi_1$, we could define two different multi-perspective configurations as shown below:

$$x = \begin{cases} P_1X & \text{for } X \in [c_1, \xi_1) \\ \widehat{P}_2X & \text{for } X \in [\xi_1, \infty) \end{cases}, \tag{2}$$

$$x = \begin{cases} \widehat{P}_2X & \text{for } X \in [c_2, \xi_1) \\ P_1X & \text{for } X \in [\xi_1, \infty) \end{cases}. \tag{3}$$

For the configuration shown in Eq. 2, the objects in front of the dolly plane are projected using the projection matrix $P_1$, while the objects beyond the dolly plane are imaged using the projection $\widehat{P}_2$ and are magnified as compared to image under projection $P_1$.

On the other hand, for the configuration shown in Eq. 3, objects in front of the dolly plane are imaged under projection $\widehat{P}_2$, while the objects beyond the dolly plane are imaged using the projection $P_1$ and hence are shrunk as compared to image under projection $\widehat{P}_2$.

The above configurations allow us to choose different projections in front and beyond a single dolly plane. As such we could define multiple dolly planes at different depths in the scene and define different projections between each consecutive pair of dolly planes. Again, we use plane-induced homographies to align the projection matrices at each dolly plane.

We now describe our general multi-perspective camera definition used for our computational zoom application. Assume we have $N-1$ dolly planes, where the $i^{th}$ dolly plane is represented by $\xi_i$. Let the dolly planes be ordered so that $\xi_1$ is the closest whereas $\xi_{N-1}$ is the farthest dolly plane. Also, assume that we have $N$ projection matrices that would be used to define different projections between each pair of consecutive dolly planes. Note that these $N$ projection matrices need not be distinguishable: this means that the same projection matrix could be used for different regions. Let $i^{th}$ projection matrix be represented by $P_i$. For our computational zoom application we assume that the normals of the dolly planes are along the positive z-axis and the pin-hole cameras used to define the multi-perspective camera definition follow approximately dolly-in motion.

We define a sequence $S$ that specifies the order in which different pin-hole projections are combined. We use camera with index $S(K)$ to define projection between the dolly planes $\xi_{K-1}$ and $\xi_K$. The projection operation for the scene points that lie between these dolly planes is given by:

$$x = \widehat{P}_K X \text{ if } X \in [\xi_{K-1}, \xi_K), \tag{4}$$

where

$$\widehat{P}_K = \prod_{i=1}^{K} H^{\xi_{i-1}}_{S(i)\to S(i-1)} P_{S(K)}. \tag{5}$$

Since $S(0)$ is not defined, $H^{\xi_0}_{S(1)\to S(0)}$, which adjusts the projection of the first camera to incorporate desired transformations such as scaling, translation, or an arbitrary homography, can be specified by the user. It can also be simply set to identity in order to use the information from the first camera directly.

## 3 PHOTOMETRIC ERROR

Here we elaborate the photometric error equation as explained in Sec. 4.1 of the main paper. The photometric error is given by:

$$E_{\text{photo}}(\pi_i(p)) = \sum_{j \in J_i(p)} \rho(N_i(p), N_j(q)), \tag{6}$$

where $\rho(\cdot)$ is a similarity function given by:

$$\rho(N_i(p), N_j(q)) = \sum_{x \in N_i(p)} w_i(x,p)\phi_{i,j}(x, H^{\pi_i(p)}_{i\to j} x). \tag{7}$$

The above summation computes patch distance between patches from images $I_i$ and $I_j$. $\phi_{i,j}$ is given by:

$$\phi_{i,j}(x,y) = (1-\alpha)\min(\|I_i(x) - I_j(y)\|, \tau_{col}) + \alpha \min(\|\nabla I_i(x) - \nabla I_j(y)\|, \tau_{grad}). \tag{8}$$

The weight function, $w_i(x,p) = \exp\left(-\frac{\|I_i(x) - I_i(p)\|}{\gamma}\right)$, acts as a soft segmentation and decreases the influence of pixels that differ a lot from the central one. We use the default values for $\tau_{col}$, $\tau_{grad}$ and $\gamma$ as defined by Galliani et al. [2015].

The outer summation aggregates different costs by matching the patch centered at $p$ in $I_i$ to patches in other images defined by set $J_i(p)$. If we knew the visibility information for the point $p$ in $I_i$, then we could define the set $J_i(p)$ for each patch accordingly and could get the most certain estimate of plane parameter $\pi_i(p)$ obtained by minimizing Eq. 6. Properly selecting such a set is very important. Selecting a very small set of images such that point $p$ in $I_i$ is visible in all selected images results in high uncertainty in plane parameter estimation, while selecting a large number of images such that point $p$ in $I_i$ is not visible in all the images may result in erroneous plane parameter estimation.

For each reference image, Galliani et al. [2015] select $M$ images to compute patch distances. This means that, for each patch centered around $p$ in image $I_i$, they compute $M$ patch distances using Eq. 7. To handle visibility they assume that patch is visible in at-least $Q \leq M$ images and hence they sort the $M$ patch distances and only aggregate least $Q$ patch distances to get the photometric error as shown in Eq. 6. Then the images belonging to the least $Q$ patch distances form our set $J_i(p)$ for the patch in current reference image.

This approach, however, could possibly give a correct estimate only for those plane parameters which are actually visible in at-least $Q$ images of the selected $M$ images. In our multi-pass approach we keep changing this parameter in each pass.

One final comment regarding the selection of $M$ images that are used to compute the set of initial patch distances with respect to patches in our reference image $I_i$. Galliani et al. [2015] select $M$ images that make large angle with respect to the reference camera. In our case of dolly-in motion, this strategy fails as the angle between viewing directions of the cameras is very small. Instead, we use a simple strategy of selecting $M$ images that are closest to the reference camera. We set a high value for $M = \min\{N/2, 15\}$ for all of the datasets used in the paper. Note that the set of selected $M$ images does not change in each pass of our multi-pass approach.

## 4 DERIVATION FOR SEGMENTATION MASKS

To synthesize images under our multi-perspective camera projection it is important to understand the fact that most of the synthesized result comes directly by sampling the rays from the source cameras. We explain this by introducing the epipole consistency criteria for our multi-perspective camera definition. Finding which multi-perspective camera rays could be sampled directly from the source cameras is trivial if we have accurate geometry. However, it is difficult to make this assumption as getting accurate geometry is very challenging. Since we are dealing with depth maps, here we explain how to find which multi-perspective camera rays could be directly sampled from the source cameras given accurate depth based image segmentation. Here we assume that we have accurate depth-based segmentation of very few source images with respect to very few dolly planes (which are used to form our multi-perspective camera). This is a more reasonable assumption as compared to availability of accurate geometry.

Let us start with a simple multi-perspective camera projection defined by two cameras $\widehat{P}_1$ and $\widehat{P}_2$ and a dolly plane. Also, let us assume the images and depth-maps corresponding to these cameras are $I_1$, $D_1$ and $I_2$, $D_2$ respectively. Without loss of generality we assume that the images cover the whole field of view of the cameras $\widehat{P}_1$ and $\widehat{P}_2$. The map $M_{i,j} = \mathbb{1}_{\{D_i \leq z_j\}}$ and its complementary $M^c_{i,j} = \mathbb{1}_{\{D_i > z_j\}}$ identify the pixels in image $i$ that are in front and beyond the dolly plane respectively. We assume that the segmentations shown by $M_{i,j}$ and $M^c_{i,j}$ are accurate. Assume $\mathbb{1}_\Omega$ represents an indicator function that represents the multi-perspective image space. Then we could represent this indicator function as:

$$\mathbb{1}_\Omega = M_{1,1} + M^c_{1,1}.$$

The desired multi-perspective camera rays that belong to pixels defined by the first term in the above equation intersect objects in front of the dolly plane and hence could be trivially sampled from the first camera. Hence $\mathcal{M}_1 = M_{1,1}$. The camera rays corresponding to pixels defined by the second term do not hit any objects in front of the dolly plane. After crossing the dolly plane these rays bend according to the projection defined by $\widehat{P}_2$. These rays could be further classified as follows:

$$\begin{aligned}\mathbb{1}_\Omega &= M_{1,1} + M^c_{1,1} \odot \mathbb{1}_\Omega \\ &= M_{1,1} + M^c_{1,1} \odot (M_{2,1} + M^c_{2,1}) \\ &= M_{1,1} + M^c_{1,1} \odot M_{2,1} + M^c_{1,1} \odot M^c_{2,1}.\end{aligned}$$

The desired rays that belong to pixels defined by $M^c_{1,1} \odot M^c_{2,1}$ do not hit any objects in front of the dolly plane, but hit objects that lie beyond the dolly plane and are visible in the second camera as

represented by $M^c_{2,1}$. These rays could be sampled trivially from the second camera and hence $\mathcal{M}_2 = M^c_{1,1} \odot M^c_{2,1}$. On the other hand the desired rays that belong to pixels defined by $M^c_{1,1} \odot M_{2,1}$ do not hit any objects in front of the dolly plane and may or may not hit some object beyond the dolly plane. This uncertainty arises because these rays are occluded, as seen by the second camera due to the objects that lie in front of dolly plane as represented by $M_{2,1}$. We deem these pixels as hole regions, which we fill by warping information from all the other images in the stack.

We could extend this analysis to the case of multi-perspective camera defined by three projection matrices and two dolly planes. $\mathcal{M}_1$ remains same as above while $\mathcal{M}_2$ will change. To find new masks for the three-cameras case, we break down the last term in above equation:

$$\begin{aligned} M^c_{1,1} \odot M^c_{2,1} &= M^c_{1,1} \odot M^c_{2,1} \odot \mathbb{1}_\Omega \\ &= M^c_{1,1} \odot M^c_{2,1} \odot (M_{2,2} + M^c_{2,2} \odot (M_{3,2} + M^c_{3,2})) \\ &= M^c_{1,1} \odot M^c_{2,1} \odot M_{2,2} + M^c_{1,1} \odot M^c_{2,1} \odot M^c_{2,2} \\ &\quad \odot (M_{3,2} + M^c_{3,2}). \end{aligned}$$

By using $M^c_{2,1} \odot M^c_{2,2} = M^c_{2,2}$, we get:

$$\begin{aligned} M^c_{1,1} \odot M^c_{2,1} = M^c_{1,1} \odot M^c_{2,1} \odot M_{2,2} + M^c_{1,1} \odot M^c_{2,2} \odot M_{3,2} \\ + M^c_{1,1} \odot M^c_{2,2} \odot M^c_{3,2}. \end{aligned}$$

Desired rays belonging to the pixels defined by $M^c_{1,1} \odot M^c_{2,1} \odot M_{2,2}$ do not hit any object in front of the first dolly plane, hit objects that lie in between first and second dolly plane and are visible in second camera. These rays could be sampled directly from the second camera. And hence $\mathcal{M}_2 = M^c_{1,1} \odot M^c_{2,1} \odot M_{2,2}$. Similarly, $\mathcal{M}_3 = M^c_{1,1} \odot M^c_{2,2} \odot M^c_{3,2}$. The regions in $\mathbb{1}_\Omega$ that are not covered by any of the $\mathcal{M}_i$'s are deemed as holes and synthesized by warping information from other regions.

For a general multi-perspective camera definition formed using $N$ cameras and $N-1$ dolly planes, the image masks are given by:

$$\mathcal{M}_i = \begin{cases} M_{1,1} & \text{for } i = 1 \\ \left( \prod_{k=1}^{i-1}{}^{\odot} M^c_{k,k} \right) \odot M^c_{i,i-1} \odot M_{i,i} & \text{for } i \in [2, N-1] \\ \left( \prod_{k=1}^{i-1}{}^{\odot} M^c_{k,k} \right) \odot M^c_{i,i-1} & \text{for } i = N, \end{cases}$$

where we use the symbol $\prod^{\odot}$ to denote the concatenation of element-wise products.

Note that the list of all the desired rays that satisfy the epipole consistency and could be sampled directly from the available source cameras is not completely covered by the set of $\mathcal{M}_i$'s. The set of $\mathcal{M}_i$'s only represent those desired rays that satisfy the epipole consistency constraints and could be found given accurate depth-based segmentation information for few images with respect to few dolly planes as shown in above equation. To find all the desired camera rays that satisfy the epipole consistency criteria, accurate geometry is required.

## REFERENCES


Silvano Galliani, Katrin Lasinger, and Konrad Schindler. 2015. Massively Parallel Multiview Stereopsis by Surface Normal Diffusion. In *IEEE CVPR*. 873–881.

R. I. Hartley and A. Zisserman. 2004. *Multiple View Geometry in Computer Vision* (second ed.). Cambridge University Press, ISBN: 0521540518.