Synthetic Aperture Focusing using a Shear-Warp Factorization of the Viewing Transform

Vaibhav Vaish* Gaurav Garg† Eino-Ville Talvala† Emilio Antunez† Bennett Wilburn†
Mark Horowitz† Marc Levoy*

*Department of Computer Science †Department of Electrical Engineering
Stanford University, Stanford, CA 94305

Abstract

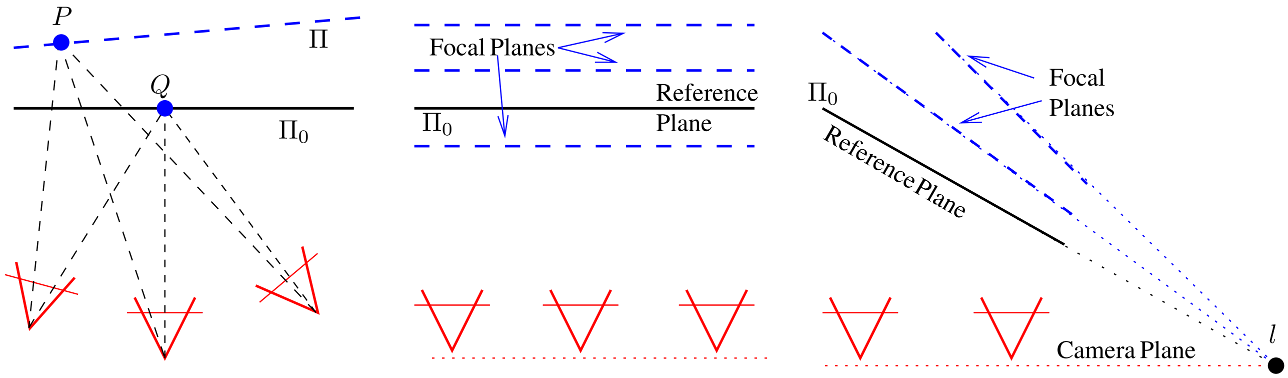
Synthetic aperture focusing consists of warping and adding together the images in a 4D light field so that objects lying on a specified surface are aligned and thus in focus, while objects lying off this surface are misaligned and hence blurred. This provides the ability to see through partial occluders such as foliage and crowds, making it a potentially powerful tool for surveillance. If the cameras lie on a plane, it has been previously shown that after an initial homography, one can move the focus through a family of planes that are parallel to the camera plane by merely shifting and adding the images. In this paper, we analyze the warps required for tilted focal planes and arbitrary camera configurations. We characterize the warps using a new rank-1 constraint that lets us focus on any plane, without having to perform a metric calibration of the cameras. We also show that there are camera configurations and families of tilted focal planes for which the warps can be factorized into an initial homography followed by shifts. This shear-warp factorization permits these tilted focal planes to be synthesized as efficiently as frontoparallel planes. Being able to vary the focus by simply shifting and adding images is relatively simple to implement in hardware and facilitates a real-time implementation. We demonstrate this using an array of 30 video-resolution cameras; initial homographies and shifts are performed on per-camera FPGAs, and additions and a final warp are performed on 3 PCs.

1. Introduction

Synthetic aperture focusing (also called dynamically reparametrized light fields) is a technique for simulating the defocus blur of a large aperture lens using multiple images of a scene, such as from a light field [4, 3]. The process consists of acquiring images of a scene from different viewpoints, projecting them onto a desired focal surface, and computing their average. In the resulting image, points that lie on the focal surface are aligned and appear sharp, whereas points off this surface are blurred out due to parallax (Fig. 1a). Researchers in computer vision and graphics have used synthetic aperture focusing to blur out occluders in front of desired focal planes, enabling them to see objects behind dense foliage [3, 9]. This ability to see behind occluders makes synthetic aperture focusing an attractive tool for surveillance.

One challenge in using synthetic aperture focusing for surveillance of dynamic scenes has been the amount of computation required. Constructing a synthetically focused image for a given focal plane requires applying a homography to each camera's image and computing their mean. The homography required for each image depends on the camera parameters and the focal plane. If we wish to change the focal plane, we need to apply different homographies to all the images. This requires substantial computation and may be difficult to achieve in real-time. However, in certain cases, we can change the focal plane without having to apply a new projective warp to the images. Consider the case when the cameras lie on a plane, and their images have been projected onto a parallel plane Π_0 (Fig. 1b). To focus on any other plane parallel to the camera plane, we need to just shift the projected images and add them [9]. In other words, we have factorized the homographies for focusing on frontoparallel planes into an initial projection (onto Π_0) followed by shifts. The initial projection needs to be applied only once.

In this paper, we explore such a factorization the case of arbitrary camera configurations and focal planes. We show that the homographies required for focusing can be factorized into an initial projection, as before, followed by a *planar homology* (a special projective warp [1]). Varying the focal plane requires varying



(a) General configuration (b) Frontoparallel Configuration (c) Scheimpflug configuration

Figure 1: (a) Synthetic aperture focusing consists of projecting camera images onto a plane Π_0 and computing their mean. Point Q on the plane Π_0 is in focus; point P not on this plane is blurred due to parallax. Projection onto a focal plane requires applying homographies to the camera images. (b) If cameras lie on a plane and their images are projected onto a parallel reference plane via homographies, then we can vary the focus through frontoparallel planes by just shifting and adding the images. This is simpler than having to apply different homographies for every focal plane. (c) We show that there exist camera configurations and families of tilted focal planes for which the focusing homographies can be decomposed into an initial projection followed by shifts. Varying the focus requires merely shifting the images, as in the frontoparallel case, plus a final warp after adding the images together.

the homologies applied to the projected images. We prove a rank-1 constraint on homology parameters to characterize homologies required for varying the focal plane. This lets us focus on any plane, for any camera configuration, without having to perform a metric calibration of the cameras; while letting a user specify focal planes in a geometrically intuitive way.

Interestingly, there are camera configurations and families of tilted focal planes for which the homologies reduce to shifts (Fig. 1c), just as in the frontoparallel case. This shear-warp factorization into an initial projection independent of the focal plane, followed by shifts to vary the focal plane, is well suited for real-time implementation in hardware. The initial projection may be implemented via a lookup table, and does not need to be changed to vary the focus. It is relatively simple to vary the focus by shifting the images in hardware. We demonstrate real-time synthetic aperture focusing with an array of 30 video cameras. The inital projection and shifts are implemented in per-camera FPGAs, and addition of images (with an optional warp applied to the final image) is done on a cluster of 3 PCs.

Our work builds upon two important concepts in multi-view geometry: the notion of plane + parallax [2, 5] which simplifies geometric analysis by projecting images from different views onto a reference plane; and the study of the space of all homologies by Zelnik-Manor et al. They show that homologies lie in a 4-D space [7]. By representing the homolgies differently, and by factoring out the epipoles we show the homology parameters actually live in a 1-D space. This helps us in specifying arbitrary focal planes.

2. Refocusing with Homologies

We will now study how to factorize the homographies required for synthetic aperture focusing into an initial projection followed by a homology. Consider an array of cameras with centers $C_0, \ldots, C_N$ whose images have been projected onto some reference plane Π_0 (Fig. 1a). Let I_i denote the projected image from the i^{th} camera. If we compute the average of the projected images I_i, we get an image focused on the reference plane. Suppose we wish to focus on a different plane, Π. One could do so by applying different homographies to the camera images. Another way would be to *reproject* each of the projected images I_i from the reference plane onto Π through center of projection C_i (Fig. 2). This reprojection is called a planar homology and can be described by a 3x3 matrix. In this section, we describe the homology matrices for different focal planes. We establish a new rank-1 constraint on homology parameters, we show how to compute the homologies without requiring metric calibration, and we enumerate the configurations in which these homologies are reduced to affine or simpler transforms.

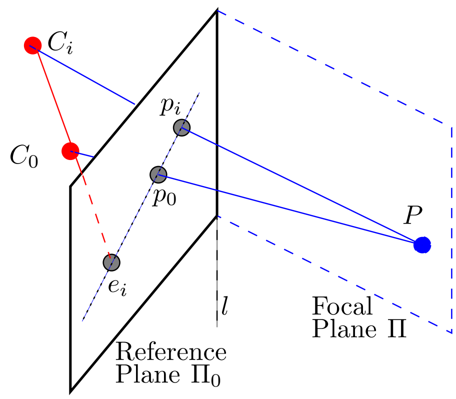


Figure 2: To change the focus from reference plane Π_0 to plane Π, we need to reproject the image from camera C_i onto Π. This projection is called a *homology.*

2.1 Planar Homologies

We begin our study of homologies by defining the coordinate systems we will use. Assume that there is a given 2D coordinate system on the reference plane. The pixels of the projected images I_i are specified in this reference coordinate system. We also need to pick a coordinate system on the new focal plane Π. To do so, we pick a reference camera - say C_0. A point P on Π will be assigned the coordinates of its projection p_0 on the reference plane through C_0 (Fig. 2). Thus, we are projecting the coordinate system on the reference plane onto the new focal plane through center of projection C_0.

Let G_i be the homology required to project I_i onto $\Pi, 1 \leq i \leq N$ ($G_0 = I$, the identity matrix). G_i maps point p_i on the reference plane to the point P on Π (Fig 2). Since P has the same coordinates as p_0, we may write $G_i p_i \cong p_0$ where G_i denotes the homology matrix, points and lines on the reference plane are represented in homogeneous coordinates and $\cong$ denotes equality up to a scale. For the ensuing analysis, it will be simpler to work with the inverse homologies $K_i = G_i^{-1}$, i.e. K_i projects points on Π onto the reference plane through center C_i and $K_i p_0 \cong p_i$.

We now proceed to characterize the 3x3 homology matrices for changing the focal plane. From projective geometry [1, 5], we know that the homology K_i can be written as

$$K_i = I + \mu_i e_i l^T$$

Here I is the identity, e_i the epipole associated with cameras C_0, C_i and the reference plane, l the line of intersection of the reference plane and Π, and μ_i is a scalar. Geometrically, varying μ_i while keeping e_i, l fixed corresponds to rotating the focal plane about axis l and moving p_i along the epipolar line through e_i and p_0. Choosing a value for μ_i amounts to choosing a focal plane (through l) for camera C_i. Suppose we are given the epipoles $e_i, 1 \leq i \leq N$. We would like to characterize $\mu_1, \ldots, \mu_N$ which are consistent, i.e. correspond to the same choice of focal plane Π. Let us call $[\mu_1 \ldots \mu_N]$ the μ-vector for homologies induced by a focal plane Π. The following result helps us characterize which vectors of $\mathbb{R}^N$ are μ-vectors for some focal plane.

Theorem *Given epipoles $e_1, \ldots, e_N$ on a reference plane as defined above, the μ-vectors for all focal planes lie in a 1-D space, i.e. μ-vectors for any two planes are equal up to scale.*

Remark: In homogeneous coordinates, points and lines can be scaled arbitrarily. The μ-vector for a focal plane will change if we change the scale of any of the epipoles or the line l. It is assumed in the theorem that we have chosen a fixed scale for each epipole e_i, this scale could be arbitrarily chosen but must be the same for all homologies we compute. If we change the scale for any of the epipoles, we will change the 1D space the μ-vectors lie in. However, as long as the scales are fixed they will still lie in a 1D space.

The proof of this theorem is presented in Appendix A.

2.2 Focusing and Calibration

Let us see how to use the preceding analysis for user-driven change of the focal plane. Suppose we know the epipoles $e_1, \ldots, e_N$ and the μ-vector $\vec{\mu} = [\mu_1 \ldots \mu_N]$ for the homologies induced by some focal plane. To specify a new focal plane, the user first chooses a line l on the reference plane (l could also be the line at infinity) through which the focal plane passes. Every focal plane through l has to have a μ-vector equal to $f\vec{\mu}$ for some scalar f. By picking a value for f, the user selects a particular focal plane through l. The homologies for this focal plane are $K_i = I + f\mu_i e_i l^T$. The synthetic aperture image with this focal plane can be computed:

$$I_{sap} = \frac{1}{N+1} \sum_{i=0}^{N} K_i^{-1} \circ I_i$$

Varying f amounts to rotating the focal plane about axis l. At $f = 0$, the focal plane coincides with the reference plane. Increasing the value of f amounts to rotating the the focal plane about l away from the

(a) (b) (c) (d)

Figure 3: User-driven change of focus. (a) An image from a light field showing a toy humvee at an angle to the plane of cameras. (b) Synthetically focused image on reference plane. Note that the left side of the humvee is out of focus, since it is not on the reference plane. (c) If we rotate the focal plane about the line indicated, we can get the full side of the humvee in focus with a tilted focal plane. (d) Plan view of our setup.

reference plane. In our system, the user can either specify f interactively by moving a slider and getting feedback from the synthetic aperture image for the corresponding focal plane, or specify a range of values of f to compute a sequence of synthetically focused images with the focal plane rotating about l (Fig. 3).

It should be clear from this discussion that to vary the focus it suffices to know the epipoles, a μ-vector for any one focal plane and the initial homographies required to project camera images onto a reference plane. These quantities may be computed using any of the projective calibration methods [1, 5] in the literature; no metric calibration (camera intrinsics or euclidean pose) is required. At the minimum, one would need to know the images of four points on a plane (for reference plane homographies) plus images of at least two points not on this plane to compute the epipoles [1] and μ-vector. For most of our experiments, the cameras (to a good approximation) do lie on a plane, and we can use the method in [9] for the necessary calibration.

2.3 Simpler Configurations

We now enumerate the cases when the homologies are not projective warps, but affine transforms or simpler. For each of these configurations, the homologies K_i lie in a proper subgroup of the group of planar homologies.[1]

General affine transforms: When the camera centers lie on a plane, and the reference plane is parallel to this plane, the epipoles $e_i = [e_i^{(x)}\ e_i^{(y)}\ 0]^T$ are points at infinity. The bottom row of the homology matrix $I + \mu e_i l^T$ becomes $[0\ 0\ 1]$. Hence, homologies for any focal plane are affine transforms. Note that this holds for arbitrary focal planes.

Scale and shift: When the focal plane and reference plane are parallel, their intersection is the line at infinity $l = [0\ 0\ 1]^T$ on the reference plane. Writing the epipoles as $e_i = [e_i^{(x)}\ e_i^{(y)}\ e_i^{(z)}]^T$, the homologies are of the form:

$$K_i = I + \mu e_i l^T = \begin{bmatrix} 1 & 0 & \mu e_i^{(x)} \\ 0 & 1 & \mu e_i^{(y)} \\ 0 & 0 & 1 + \mu e_i^{(z)} \end{bmatrix}$$

This is just a scale followed by a shift. Thus, if we wish to vary the focus through a family of planes parallel to the reference plane, we need to just scale and shift the images before computing their average. Note that this holds for arbitrary camera positions.

Shifts (Scheimpflug Configuration): Consider the case when the cameras lie on a plane, and the camera plane, reference plane and desired focal plane intersect in the same line l (Fig 1c). The epipoles lie on on this line l. Suppose we redefine the cooordinate system on the reference plane so that $l \cong [0\ 0\ 1]^T$. This combines the previous two conditions, and the homologies are reduced to shifts:

$$K_i = I + \mu \begin{bmatrix} e_i^{(x)} \\ e_i^{(y)} \\ 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \mu e_i^{(x)} \\ 0 & 1 & \mu e_i^{(y)} \\ 0 & 0 & 1 \end{bmatrix}$$

This condition is analogous to the Scheimpflug condition in photography, which is required to focus on tilted planes. Specifically, the lens plane, sensor plane and focal plane must intersect in a common line [6]. This case is well suited for a real-time implementation in hardware as the image shifts required to

[1] Unless otherwise stated, we will assume we have established an affine coordinate system on the reference plane.

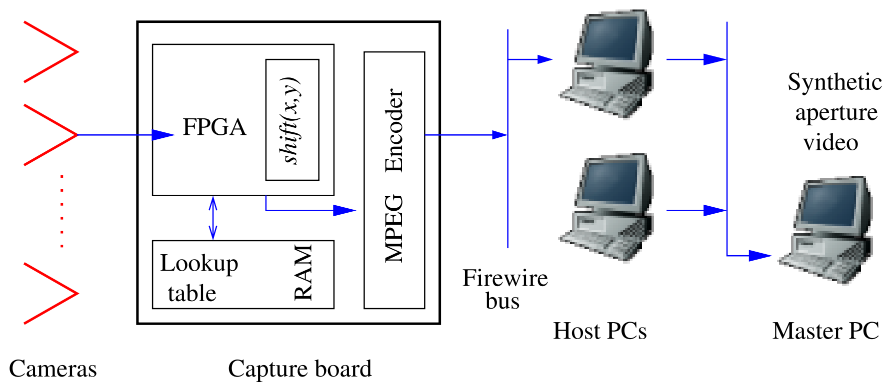


Figure 4: Real-time system. Each video stream goes to a capture board where a homography is applied (using a lookup table) followed by a shift (x, y). The MPEG-compressed video is sent over firewire to a host PC, where warped frames from different cameras are added. A master PC adds the streams from all host PCs, displaying synthetic aperture video. The focal plane can be changed by varying the shifts.

vary the focal plane are easy to realize in hardware. This generalizes the frontoparallel case studied in [9] (Fig. 1b). After shifting and adding the images, we can warp the resulting image back to the original coordinate system on the reference plane if desired.

In fact, all configurations for which varying the focus requires only shifts have to be Scheimpflug configurations. $K_i = I + \mu e_i l^T$ is a translation only if $l = [0\ 0\ 1]^T$ and $e_i^{(z)} = 0$. This means the epipoles lie on l, i.e. the camera plane intersects the reference plane in the same line l as the focal plane.

3. Real-time Synthetic Focus

We have implemented synthetic aperture video on an array of 30 cameras in our laboratory. Our camera array is based on the architecture described in [8], with a video resolution of 320x240 grayscale, 30 frames/sec. Here we will concentrate on how the shear-warp factorization lets us vary the focal plane in real-time.

The processing pipeline is shown in Fig 4. The video stream from each camera is sent to its capture board. The FPGA applies the initial homography required for projection onto the reference plane using a precomputed lookup table stored in RAM. The warped frames are shifted, MPEG compressed and transmitted to the host PC. Each host PC receives streams from 15 cameras, which are decoded, added and sent over the network to a master PC. The master PC adds the streams from the hosts and displays the final synthetic aperture video. It also warps the final image back to the original coordinate system of the reference plane, if desired, as described in the previous section.

In general, varying the focal plane requires changing the homographies being applied in the FPGAs. However, loading new lookup tables into the RAM takes about 30 seconds; we cannot change the focus interactively this way. This is where the shear-warp factorization is useful. For the Scheimpflug configuration, the homographies can be factored into a reference plane homography followed by a shift. Changing the focal plane through the family of planes described in the previous section only requires changing the shifts in the FPGAs. This is easy to do in real-time. In our interactive system, the focal plane is changed by having the user move a slider, which updates the shifts in the FPGAs.

Instead of using a lookup table, one could try to implement projective warps by multiplying each pixel coordinate with a 3x3 homography matrix and finding the intensity values at the resulting point in the captured frame (backward warp). This avoids the expense of changing a large lookup table for changing the focus. However, this approach requires multipliers and at least one divide per pixel. Implementing these in hardware would use a substantial fraction of our FPGA, and might not fit at all given more modest per-camera hardware. In contrast, using a lookup table is relatively simple to realize in the FPGAs, and could let us apply arbitrary warps which would be necessary anyway if we wished to correct for lens distortion or have non-planar focal surfaces. The drawback is that it constrains the ways in which we can vary the focus.

3.1 Results

We show two examples of our system in action. The first scene consists of a person moving forward towards the cameras, with people walking in front of him (Fig. 6). By adjusting the focal plane with a slider as the subject moves, the user is able to keep him in focus while the people occluding him are blurred out.

The second scene consists of a suitcase placed at an angle to the camera array (Fig. 7). Although it is not possible to bring the entire suitcase into good focus using frontoparallel planes, using tilted focal planes from a Scheimpflug configuration, we can bring the entire suitcase into focus, while people in front of it are blurred out. (We urge the reader to view a video demonstrating the capabilities of our system in action at *http://graphics.stanford.edu/papers/shear-warp/a3diss.avi*).

4. Conclusions

In this paper, we have shown the homographies required for synthetic aperture focusing on arbitrary focal planes can be factorized into an initial projection followed by a homology. We have categorized the camera and focal plane configurations for which homologies are affine or simpler warps. For cameras and focal planes in the Scheimpflug configuration, these homologies are reduced to shifts, facilitating a hardware implementation in which we can change the focal plane in real-time. Given the ability of synthetic aperture imaging to see around occluders, we feel this system would be useful for surveillance and reconaissance. Our analysis also shows how to implement synthetic aperture focusing without having to perform a metric calibration.

The main limitation of the system is that we are restricted to a family of focal planes that pass through a line (or parallel focal planes, if this line is at infinity). To change this family - for example, to switch from frontoparallel focal planes to tilted planes - we need to update the lookup tables for all our cameras, which takes about 30 seconds.

We would like to extend our system to work with more cameras and handle non-planar focal surfaces. An interesting challenge would be to automatically track moving objects in real-time, even if they get occluded in most of the cameras. In our experiments with manually tracking people through crowds, we learnt that it is difficult to have the tracked person in perfect focus. Thus, computer-assisted focusing would be a desirable addition to our system.

Acknowledgements: We wish to thank Augusto Román, Billy Chen and Abhishek Bapna for assisting in acquisitions. This work was partially supported by grants NSF IIS-0219856-001, DARPA NBCH 1030009, and the Reed-Hodgson Stanford Graduate Fellowship.

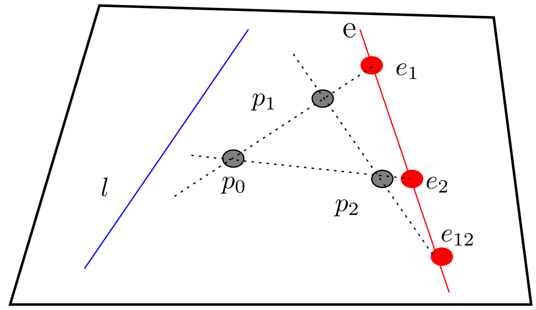


Figure 5: p_0, p_1, p_2 are images of a point P on the focal plane Π in cameras C_0, C_1, C_2 (see Fig 1b). e_1, e_2, e_{12} are the epipoles. The plane through camera centers C_0, C_1, C_2 intersects the reference plane in a line e containing these epipoles. l is the intersection of the focal plane with the reference plane.

References

[1] R. Hartley and A. Zisserman. Multiple View Geometry, *Cambridge University Press*, 2000.

[2] M. Irani, P. Anandan and D. Weinshall. From Reference Frames to Reference Planes: Multi-View Parallax Geometry and Applications. In *Proc. of ECCV*, 1996.

[3] A. Isaksen, L. McMillan and S. Gortler. Dynamically Reparametrized Light Fields. In *Proc. of ACM SIGGRAPH*, 2000.

[4] M. Levoy and P. Hanrahan. Light Field Rendering. In *Proc. of ACM SIGGRAPH*, 1996.

[5] B. Triggs. Plane + Parallax, Tensors and Factorization. In *Proc. of ECCV*, 2000.

[6] R. Wheeler. Notes on Viewcam Geometry. *http://www.bobwheeler.com/photo/ViewCam.pdf*

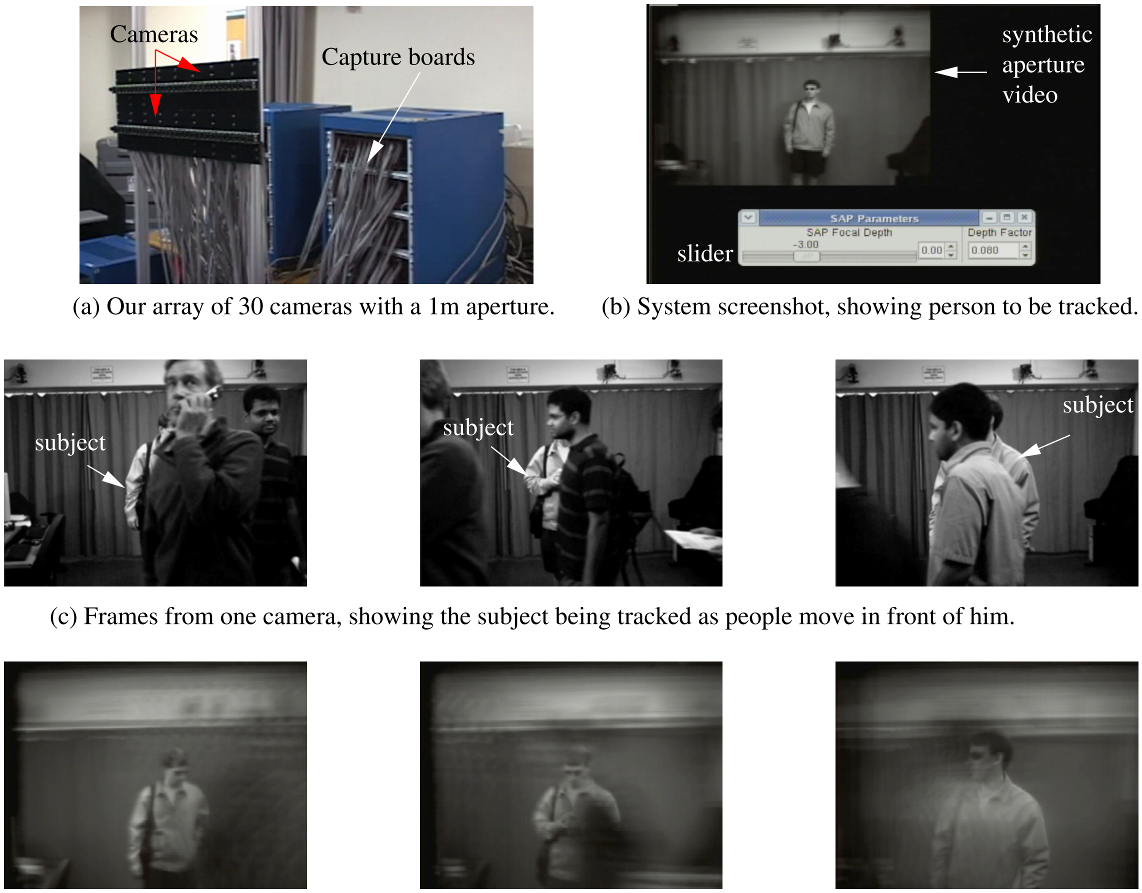
[7] L. Zelnik-Manor and M. Irani. Multi-view Subspace Constraints on Homographies. In *Proc. of IEEE ICCV,* 1999.

[8] B. Wilburn. High Performance Imaging Using Arrays of Inexpensive Cameras. *PhD Thesis*, 2004.

[9] V. Vaish, B. Wilburn, N. Joshi, M. Levoy. Using Plane + Parallax to Calibrate Dense Camera Arrays. In *Proc. of IEEE CVPR,* 2004.

Appendix A

Here we prove the theorem stated in Section 2.1. To show the μ-vectors live in a 1D space, it suffices



(a) Our array of 30 cameras with a 1m aperture.

(b) System screenshot, showing person to be tracked.

(c) Frames from one camera, showing the subject being tracked as people move in front of him.

(d) Corresponding frames from the synthetic aperture video. The focal plane follows the subject as he moves forward.

Figure 6: Real-time synthetic aperture video, used to track a person moving through a crowd of people.

to show that the ratio $\mu_i/\mu_j, 1 \leq i < j \leq N$ is independent of the plane Π.

Without loss of generality, we may take $i = 1, j = 2$. Let P be a point on Π, and p_0, p_1, p_2 be its projections onto the reference plane through centers of projection C_0, C_1, C_2 (Fig. 5). Let e_{12} be the epipole associated with camera centers C_1, C_2 lying on the reference plane. The epipoles e_1, e_2, e_{12} lie on the line of intersection of the reference plane and the plane of camera centers C_0, C_1, C_2 (by Desargues' theorem). By epipolar geometry, e_{12}, p_1, p_2 are collinear. Let $|a\ b\ c|$ denote the determinant of the matrix whose columns are the 3-vectors a, b, c, and note that $|a\ \ b\ \ c| = 0$ if a, b, c are collinear. We have

$$\begin{aligned} p_1 \cong K_1 p_0 &= p_0 + \mu_1 e_1 (l^T p_0) && (1) \\ p_2 \cong K_2 p_0 &= p_0 + \mu_2 e_2 (l^T p_0) && (2) \\ |e_{12}\ e_1\ e_2| &= 0 && (3) \\ |e_{12}\ p_1\ p_2| &= 0 && (4) \end{aligned}$$

where the first two relations follow from the homologies K_1, K_2 and the last two from collinearity. If we substitute (1) and (2) in (4), expand using properties of determinants and (3), we get

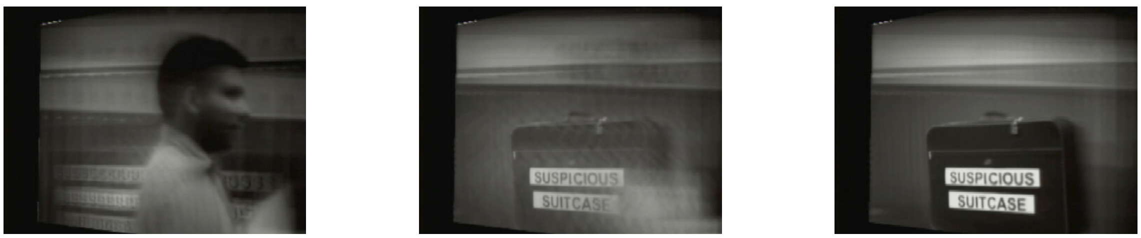
$$\begin{aligned} 0 &= |e_{12}\ p_1\ p_2| \\ &= |e_{12}\ p_0 + \mu_1 e_1 (l^T p_0)\ p_0 + \mu_2 e_2 (l^T p_0)| \\ &= |e_{12}\ \mu_1 e_1 (l^T p_0)\ p_0| + |e_{12}\ p_0\ \mu_2 e_2 (l^T p_0)| \\ &= \mu_1 |e_{12}\ e_1\ p_0| + \mu_2 |e_{12}\ p_0\ e_2| \\ &= \mu_1 |e_{12}\ e_1\ p_0| - \mu_2 |e_{12}\ e_2\ p_0| \end{aligned}$$

This yields

$$\mu_1/\mu_2 = |e_{12}\ e_2\ p_0| / |e_{12}\ e_1\ p_0|$$

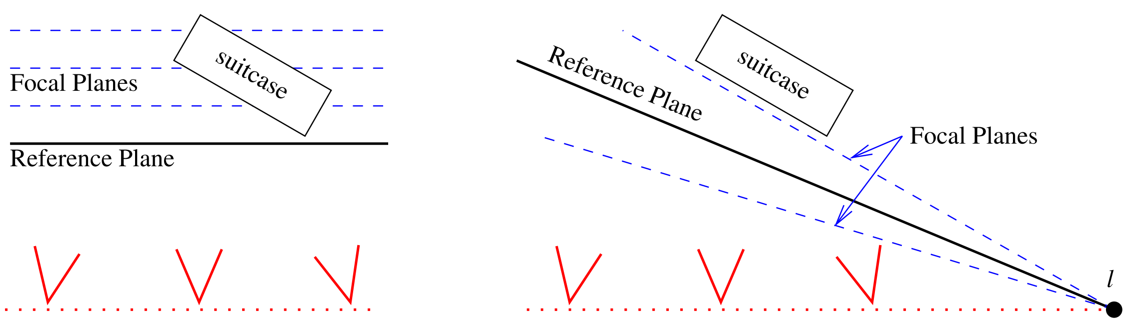
The right hand side does not depend on the plane Π (we can also show it does not depend on p_0). This completes the proof.

(a) Frames from one camera showing a person moving in front of a suitcase, placed at an angle to the plane of cameras.



(b) Corresponding frames from synthetic aperture video using tilted focal planes. In the left image, the focal plane passes through the person in front. In the remaining two images, the focus is on the suitcase.

(c) Synthetic aperture video using frontoparallel focal planes can get only part of the suitcase in good focus.



(d) Plan view. Left: frontoparallel planes, as in (c). Right: tilted planes in Scheimpflug configuration as in (b).

Figure 7: Focusing with tilted planes.