A. Computational Complexity of Multi-head Cost Volume

Suppose the dimension of an input activation is CHW, and the max disparity is d. The number of multiply-accumulate operations (#MAC) of the original cost volume is $3CHWd$ (please refer to Algorithm 1). The layer norm along channel dimension is defined as Eq. (10), whose #MAC is $3HWd$. Replacing the cosine similarity with the dot product, and adding the layer norm before the loop reduce the #MAC. The #MAC of multi-head cost volume is $2 \times 3CHW + CHWd < 3CHWd$. (The definition of parameters in Eq. (10) follow [4].)

$$y = \frac{x - E[x]}{\sqrt{Var[x] + \epsilon}}\gamma + \beta \tag{10}$$

B. SIFT v.s. Conv Network

As [30] analyses, the computational complexity of SIFT for an $N \times N$ image with $n \times n$ tiles is

$$\Theta\left(\frac{(n+x)^2}{p_i \Gamma_0} + \alpha\beta N^2 x^2 \Gamma_1 + \frac{(\alpha\beta+\gamma)n^2 \log x}{p_o \Gamma_2}\right) \tag{11}$$

where x is the neighborhood of tiles. $N \gg n > x$ in most cases, we can simplify the complexity as $O(N^2)$.

For convolutional networks, the computational complexity for an $N \times N$ input activation with C channels in input and C' channels in output is,

$$O(N^2 C C' k^2) \tag{12}$$

where k is the convolution kernel size. $N \gg C > k$ in most cases, we can also simplify the complexity as $O(N^2)$.

Based on the above analysis, we can conclude that: **Supervised learning convolutional neural networks capable of the same task will not perform worse efficiency for all computer vision algorithms requiring key point matching.** As the same, CNN is as efficient as, or even outperforms, classic algorithms in homography estimation tasks.

In practice, many optimizations for CNNs have been proposed, and CNN computations are more hardware-friendly. In contrast, [30] has been proven that without improvements in the input bandwidth, the power of multicore processing cannot be used efficiently for SIFT. Therefore, CNNs are generally a more efficient approach. Based on the report in [31] and our experiments, keypoint matching takes around 300ms on both smartphones and laptops. There is no significant speed up from smartphone to laptop, showing the limitations of keypoint matching.

C. Dataset Setting

DTU setting: Based on previous implementations and common practices [33], we selected the evaluation set as scans {1, 4, 9, 10, 11, 12, 13, 15, 23, 24, 29, 32, 33, 34, 48, 49, 62, 75, 77, 110, 114, 118}, validation set: scans {3, 5, 17, 21, 28, 35, 37, 38, 40, 43, 56, 59, 66, 67, 82, 86, 106, 117}, and the rest is training set.

L	1	2	3	4	5	7	8	9	10	11	12	12
R	2	3	4	5	6	6	7	8	9	10	11	13
L	13	14	15	16	17	18	19	21	22	23	24	25
R	14	15	16	17	18	19	20	20	21	22	23	24
L	26	27	28	29	29	30	31	32	33	34	35	36
R	25	26	27	28	30	31	32	33	34	35	36	37
L	38	40	41	42	43	44	45	45	46	47	48	49
R	39	39	40	41	42	43	44	44	45	46	47	48

Table 6. Input images pairs of DTU Dataset

Additionally, our network takes two images as input, but the depth map is aligned with the left image. This means that the left and right inputs cannot be interchanged. We need to match the stereo images to ensure that the relative position of the left input is indeed on the left side of the right input. The image match list is shown as Tab. 6.

ADT setting: As mentioned, ATD ignores users' bodies in the images when rendering ground truth depth maps which causes inconsistencies between the input images and the predicted results. We selected subsets of the scene where no other users were present. The selected subset can be obtained by this query link: https://explorer.projectaria.com/adt?q=%22is_multi_person+%3D%3D+false%22

D. Metrics Definition

Here are the definitions of the metrics we used for evaluation:

$$AbsRel = \frac{1}{N}\sum_{i=1}^{N}\frac{\hat{y}_i - y_i}{y_i} \tag{13}$$

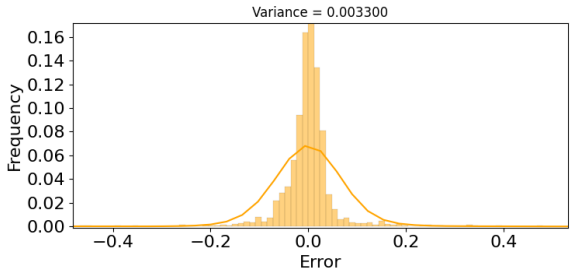
$$D1 = \frac{1}{N}\sum_{i=1}^{N}\mathbf{1}\left(\frac{|\hat{y}_i - y_i|}{y_i} \leq 5\%\right) \tag{14}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2} \tag{15}$$

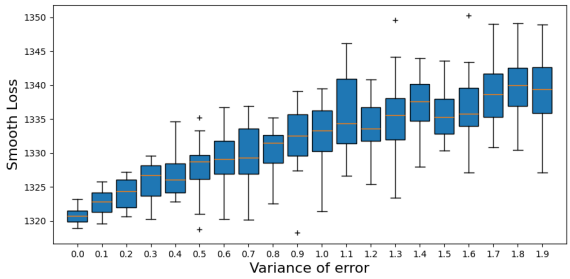
where N is the total number of pixels of the depth maps, y_i is the value of pixels in ground truth map, and $\hat{y}_i$ is the value of pixels in prediction.

E. Robustness Analysis

HomoDepth is a muti-task learning structure, and the depth estimation depends on the predicted homography matrix. Thus, the accuracy of homography estimation affects depth estimation. In this section, we address two questions:



(a) **The statistics of the error in homography estimation.** The curve represents the Gaussian fitting of the statistical results.



(b) **The simulation of the influence of noise added to the homography matrix.** The blue bars represent the smooth loss error ranges corresponding to noise variances σ, and the cross points represent outliers.

Figure 9. The robustness analyze of HOMODEPTH

(1) How sensitive is the depth estimation to the homography estimation? (2) How stable is the homography estimation?

Homography Estimation Errors. We analyze the error of HOMODEPTH during homography estimation. In practice, the error variance is as small as 0.003, as shown in Fig. 9a. This indicates that the homography estimation of HOMODEPTH is highly accurate.

Sensitivity Study. In HOMODEPTH, we inject noise $n \sim N(0, \sigma)$, where $\sigma \in [0, 2]$, to the elements of estimated homography matrix before it is passed to the multi-head cost volume blocks. Then, we investigate the final depth estimation errors. The instability is quantified by examining the noise variance and corresponding changes in the smooth loss function Eq. (6) corresponding to depth estimation. As shown in Fig. 9b, the standard deviation trends indicate that depth estimation remains stable when $\sigma < 0.5$. The linear fitting demonstrates that the loss values increase as the noise variance grows.

F. Application Scenarios

For one-shot scenarios, we recommend using HOMODEPTH. For continuous frames scenarios, we assume that the relative positions of the cameras on AR glasses remain stable over a short period. Therefore, we recommend first running HOMODEPTH to obtain depth estimation while simultaneously deriving the homography between the two cameras. For subsequent stereo inputs, rectification can be quickly applied with homography, and MULTIHEADDEPTH can be utilized to achieve higher efficiency.